

CUBESAT ASTRONOMY MISSION MODELING USING THE HORIZON  
SIMULATION FRAMEWORK

A Thesis  
presented to  
the Faculty of California Polytechnic State University,  
San Luis Obispo

In Partial Fulfillment  
of the Requirements for the Degree  
Master of Science in Aerospace Engineering

by  
Alexander Johnson  
September 2019

© 2019  
Alexander Johnson  
ALL RIGHTS RESERVED

## COMMITTEE MEMBERSHIP

TITLE: CubeSat Astronomy Mission Modeling Using the Horizon Simulation Framework

AUTHOR: Alexander Johnson

DATE SUBMITTED: September 2019

COMMITTEE CHAIR: Eric Mehiel, Ph.D.  
Interim Associate Dean of Undergraduate Affairs

COMMITTEE MEMBER: Kira Abercromby, Ph.D.  
Professor of Aerospace Engineering

COMMITTEE MEMBER: Amelia Greig, Ph.D.  
Assistant Professor of Aerospace Engineering

COMMITTEE MEMBER: Russell Genet, Ph.D.  
Research Scholar in Residence

## ABSTRACT

### CubeSat Astronomy Mission Modeling Using the Horizon Simulation Framework

Alexander Johnson

The CubeSat Astronomy Network is a proposed system of multiple CubeSat spacecraft capable of performing follow-up observations of astronomical targets of interest. The system is intended to serve as a space-borne platform that can complement existing systems utilized for astronomical research by undergraduate and high school students. Much research and development work has been performed to develop model-based system engineering methodologies and products for CubeSat missions, including the Horizon Simulation Framework. The Horizon Simulation Framework enables the development of system models using the Extended Markup Language (XML), and its simulation program can generate system simulations over model-specified time-spans. System requirements and constraints, as well as subsystem dependencies and functions, can also be directly specified in these models. Previous work using the framework has been performed to characterize day-in-the-life operations for Earth-observing spacecraft. A similar goal is intended for modeling the CubeSat Astronomy Network: simulating mission operations during nominal conditions to validate system and subsystem requirements. By developing this model, system and subsystem requirements derived in the course of preliminary design for the Network can be analyzed, modelled, and evaluated for feasibility. These results can then be used to inform design decisions related to system architecture and concept of operations at the early stages of design, while the models themselves can grow and mature alongside project development and be re-used for future design work.



## ACKNOWLEDGMENTS

Thanks to Dr. Russ Genet and Dave Rowe, for their guidance and insight, and for sharing with me the dream of accessible space-based astronomy. Thanks also to PlaneWave Instruments, for making possible the initial research that drove the objectives of this work, and the Alt-Az Initiative for hosting me during their summer workshop in 2018. Thanks also to Joseph Carpico, for introducing me to this endeavor, and sharing his professional experience and wisdom these past two years. Special thanks to Dr. Eric Mehiel, for his mentorship, his patience, and providing the most rewarding opportunity to put my learning into practice. Thanks to my committee members Dr. Kira Abercromby and Dr. Amelia Greig, for their time so generously donated, and their advice and guidance throughout my years as an undergraduate.

Special thanks to Marlene and Wesley Anderson, for helping to teach their grandson to become a man proud of where he stands today because of them. Special thanks to Fritz Whitney, for his tremendous compassion and his earnest friendship. Special thanks to Will and Emma, for keeping me creative, spontaneous, humble, and, above all else, their brother. Special thanks to Kelly Tompkins, for her ceaseless support and kindness, which can neither be overestimated nor neglected.

Thanks, finally, to Jeffrey and Malia, for raising a son that considers himself blessed to have been raised, taught, helped, and inspired by the both of you.

# TABLE OF CONTENTS

	Page
LIST OF TABLES . . . . .	viii
LIST OF FIGURES . . . . .	ix
CHAPTER	
1 Introduction . . . . .	1
1.1 Summary of Objectives . . . . .	1
1.2 Organization of Thesis . . . . .	2
1.3 Thesis Statement . . . . .	2
2 Background . . . . .	4
2.1 Survey of Undergraduate Astronomy . . . . .	4
2.2 Model-Based Systems Engineering Methodologies . . . . .	6
2.2.1 JPL State Analysis . . . . .	7
2.2.2 Extended Applications Beyond Conceptual Design . . . . .	8
2.3 CubeSat Model-Based Systems Engineering . . . . .	9
2.3.1 CubeSat MBSE Applications . . . . .	10
2.4 CubeSat and Nanosatellite Astronomy Heritage and Development . .	11
2.4.1 Previous Nanosatellite Astronomy Missions . . . . .	12
2.4.2 Current Development of Nanosatellite Astronomy Missions . .	14
2.5 CubeSat Astronomy Network . . . . .	15
2.5.1 System Architecture . . . . .	16
2.5.2 Day-in-the-Life Concept of Operations . . . . .	17
2.5.3 Key and Driving Requirements . . . . .	19
2.5.4 Bus Design . . . . .	21
2.6 Horizon Simulation Framework . . . . .	21
2.6.1 Glossary of Horizon Terminology . . . . .	22
2.6.2 Architecture . . . . .	23
2.6.3 Scheduling Algorithm . . . . .	24
2.6.4 Modeling Segment . . . . .	26
2.6.5 Applicable Prior Work . . . . .	28
3 Methodology . . . . .	31
3.1 Reference Frames . . . . .	31
3.1.1 Earth-Centered Inertial Frame . . . . .	32
3.1.2 Local Horizontal Frame . . . . .	32
3.1.3 Body Frame . . . . .	34
3.2 Model Subsystem Architecture . . . . .	35
3.3 Dynamics Models . . . . .	36
3.4 ADCS Subsystem Model . . . . .	43
3.4.1 Linearized PD Attitude Controller . . . . .	44
3.4.2 LQR Attitude Controller . . . . .	47
3.4.3 Reaction Wheel Momentum Desaturation Controller . . . . .	50
3.4.4 Commanded Attitude Pointing Modes . . . . .	51
3.5 Payload Subsystem Model . . . . .	53

3.6	Mission Data Handling Subsystem Model . . . . .	54
3.7	Communications Subsystem Model . . . . .	55
3.8	Power Subsystem Model . . . . .	56
3.9	Design Reference Mission . . . . .	57
3.9.1	Single Satellite Reference Mission . . . . .	59
3.9.2	Constellation Reference Mission . . . . .	59
4	Analysis . . . . .	62
4.1	Summary of Simulation Results . . . . .	62
4.2	Single Satellite Case Results . . . . .	62
4.3	Constellation Case Results . . . . .	77
4.3.1	Constellation Attitude Results . . . . .	77
4.3.2	Constellation Reaction Wheel Results . . . . .	79
4.3.3	Constellation Science Data Results . . . . .	79
4.3.4	Constellation Power Generation Results . . . . .	80
5	Results . . . . .	81
5.1	Summary of Results . . . . .	81
5.2	Lessons Learned . . . . .	82
5.2.1	Concurrent Development Practices . . . . .	83
5.2.2	Model and System Architectures: Similarities and Differences . . . . .	84
6	Conclusion . . . . .	87
6.1	Summary of Simulated Results . . . . .	87
6.2	Summary of Design Assessment . . . . .	87
6.3	Summary of Necessary Design Changes and Analysis . . . . .	88
6.4	Summary of Assessment of Horizon Simulation Framework . . . . .	88
6.5	Future Work . . . . .	89
6.5.1	Interoperability with CubeSat Reference Model . . . . .	90
6.5.2	Interoperability with Design and Analysis Software . . . . .	90
6.5.3	Additional Functionality in Horizon . . . . .	91
6.5.4	Next Steps in CubeSat Astronomy Network System Design . . . . .	92
	BIBLIOGRAPHY . . . . .	93
	APPENDICES	
A	Constellation State Data . . . . .	101
B	Exponential Atmosphere Model . . . . .	117
C	World Magnetic Model . . . . .	118
D	Geopotential Disturbances . . . . .	123
D.1	Geopotential Harmonics . . . . .	123
D.2	Third Order Zonal Harmonic . . . . .	124
D.3	Fourth Order Zonal Harmonic . . . . .	125
D.4	Fifth Order Zonal Harmonic . . . . .	125
D.5	Zonal Harmonic Coefficients . . . . .	125
E	Simulated Conceptual Link Budget Summary . . . . .	127
F	Simulated Conceptual Power Budget . . . . .	129

## LIST OF TABLES

Table		Page
2.1	Telescope Conceptual Design Key Parameters . . . . .	30
3.1	Table of System Tasks and Targets . . . . .	35
3.2	Conceptual Design Mass Properties . . . . .	38
3.3	Blue Canyon XACT-15 Module Parameters . . . . .	44
3.4	RWP-015 Key Parameters . . . . .	45
3.5	LQR Controller Step Response Parameters . . . . .	50
3.6	Mission Data Handling Subsystem Parameters . . . . .	55
3.7	Communication Subsystem Key Parameters . . . . .	55
3.8	Power Subsystem Key Parameters . . . . .	57
3.9	CubeSat Astronomy Network Reference Mission Targets . . . . .	58
3.10	Single Satellite Orbit Initial Conditions . . . . .	59
3.11	Constellation Reference Orbit Initial Conditions . . . . .	61
4.1	Reaction Wheel Saturation Trend Analysis . . . . .	68
4.2	Reaction Wheel Saturation Trend Analysis, Continued . . . . .	69
4.3	Data Storage Utilization Trend Analysis . . . . .	72
B.1	Exponential Atmosphere Model Coefficients . . . . .	117
C.1	WMM Coefficients A . . . . .	118
C.2	WMM Coefficients B . . . . .	119
C.3	WMM Coefficients C . . . . .	120
C.4	WMM Coefficients D . . . . .	121
C.5	WMM Coefficients E . . . . .	122
D.1	Zonal Harmonic Coefficients Up To Fifth Order . . . . .	126
F.1	Conceptual Power Budget . . . . .	129

## LIST OF FIGURES

Figure		Page
2.1	JPL State Analysis Control Architecture[1] . . . . .	7
2.2	BRITE Spacecraft[2] . . . . .	13
2.3	ASTERIA 6U CubeSat Spacecraft[3] . . . . .	14
2.4	CubeSat Astronomy Network Concept of Operations . . . . .	18
2.5	Telescope Conceptual Design[4] . . . . .	21
2.6	Example Horizon Dependency Diagram . . . . .	27
3.1	ECI Reference Frame . . . . .	32
3.2	Local Horizontal Reference Frame . . . . .	33
3.3	Telescope Body Axes and Approximate Center of Mass . . . . .	35
3.4	Asset Subsystem Dependencies for the CubeSat Astronomy Network	37
3.5	Surface Element Geometry for Drag Calculation . . . . .	40
3.6	Idle and Desaturation Pointing Mode . . . . .	52
3.7	Imaging Pointing Mode . . . . .	53
3.8	Ground Station Communication Pointing Mode . . . . .	53
3.9	CubeSat Astronomy Network Constellation Configuration . . . . .	60
4.1	Single Satellite Mission Orbit Geometry . . . . .	63
4.2	Single Satellite Mission Attitude Time Series . . . . .	64
4.3	Single Satellite Mission Pointing Error Time Series . . . . .	65
4.4	Single Satellite Mission Body Rates Time Series . . . . .	66
4.5	Single Satellite Mission Wheel Speeds Time Series . . . . .	67
4.6	Single Satellite Wheel Trends Time Series . . . . .	68
4.7	Single Satellite Wheel Desaturation Time Series and Trend . . . . .	69
4.8	Single Satellite Mission Command Torque Time Series . . . . .	70
4.9	Single Satellite Mission Sun Angle Time Series . . . . .	71
4.10	Single Satellite Mission Ram Angle Time Series . . . . .	72
4.11	Single Satellite Mission Nadir Angle Time Series . . . . .	73
4.12	Single Satellite Mission Raw Bytes Recorded Time Series . . . . .	74
4.13	Single Satellite Mission Data Storage Utilized Time Series . . . . .	75
4.14	Mission Data Storage Utilized Trend Time Series . . . . .	76
4.15	Downlink Data Storage Trend Time Series . . . . .	77
4.16	Single Satellite Mission Solar Power Input Time Series . . . . .	78
4.17	Constellation Payload Capture Time Series . . . . .	80
A.1	Constellation Attitude Time Series for Satellite 1 . . . . .	101
A.2	Constellation Attitude Time Series for Satellite 2 . . . . .	102
A.3	Constellation Attitude Time Series for Satellite 3 . . . . .	102
A.4	Constellation Body Rate Time Series for Satellite 1 . . . . .	103
A.5	Constellation Body Rate Time Series for Satellite 2 . . . . .	103
A.6	Constellation Body Rate Time Series for Satellite 3 . . . . .	104

A.7	Constellation Sun Angle Time Series for Satellite 1 . . . . .	104
A.8	Constellation Sun Angle Time Series for Satellite 2 . . . . .	105
A.9	Constellation Sun Angle Time Series for Satellite 3 . . . . .	105
A.10	Constellation Ram Angle Time Series for Satellite 1 . . . . .	106
A.11	Constellation Ram Angle Time Series for Satellite 2 . . . . .	106
A.12	Constellation Ram Angle Time Series for Satellite 3 . . . . .	107
A.13	Constellation Nadir Angle Time Series for Satellite 1 . . . . .	107
A.14	Constellation Nadir Angle Time Series for Satellite 2 . . . . .	108
A.15	Constellation Nadir Angle Time Series for Satellite 3 . . . . .	108
A.16	Constellation Wheel Speed Time Series for Satellite 1 . . . . .	109
A.17	Constellation Wheel Speed Time Series for Satellite 2 . . . . .	109
A.18	Constellation Wheel Speed Time Series for Satellite 3 . . . . .	110
A.19	Constellation Command Torque Series for Satellite 1 . . . . .	110
A.20	Constellation Command Torque Series for Satellite 2 . . . . .	111
A.21	Constellation Command Torque Series for Satellite 3 . . . . .	111
A.22	Constellation Science Data Recorded Time Series for Satellite 1 . .	112
A.23	Constellation Science Data Recorded Time Series for Satellite 2 . .	112
A.24	Constellation Science Data Recorded Time Series for Satellite 3 . .	113
A.25	Constellation Data Storage Time Series for Satellite 1 . . . . .	113
A.26	Constellation Data Storage Time Series for Satellite 2 . . . . .	114
A.27	Constellation Data Storage Time Series for Satellite 3 . . . . .	114
A.28	Constellation Power Generation Time Series for Satellite 1 . . . . .	115
A.29	Constellation Power Generation Time Series for Satellite 2 . . . . .	115
A.30	Constellation Power Generation Time Series for Satellite 3 . . . . .	116
E.1	Simulated Transmitter Downlink Budget . . . . .	127
E.2	Simulated Receiver Downlink Budget . . . . .	128

# Chapter 1

## INTRODUCTION

### 1.1 Summary of Objectives

To investigate and characterize the feasibility of leveraging the Horizon Simulation Framework for systems engineering activities related to the development of the CubeSat Astronomy Network, models were developed for assessing the performance and validating the system-level requirements of the CubeSat Astronomy Network. Prior design work performed in assessing the feasibility of utilizing CubeSat spacecraft for performing astronomical science data collection was used to inform much of the models development and integration. These models were used to simulate the performance of the network during a typical use case: continuous time-series observation of an astronomical science target.

The results of this endeavor are presented and analyzed, with the following general conclusions: the Framework can be leveraged for use in model-based systems engineering activities, yet may require further maturation to match the capabilities of other modeling frameworks; interoperability enabled by the .NET Framework with other software packages and resources may be leveraged to enable the utilization of free and open-source software that can extend its existing capabilities; and the system-level requirements of the CubeSat Astronomy Network, including the capability of performing continuously-sampled time-series observations, may possibly be feasibly fulfilled by CubeSat-format spacecraft and hardware, pending further design activities, reviews, and securement of funding.

## **1.2 Organization of Thesis**

This thesis is organized into six major sections, in the following order: the first section consists of introductory material, summarizing the research, methodology, and results of this thesis; the second section details background information pertaining to the research performed and information regarding the topics pursuant to the review of relevant literature; the third section details the analytical and computational methods utilized to produce the relevant data for analysis and characterization; the fourth section presents the results of the previously described methodology; the fifth section presents the analytical findings that resulted from characterization of the resultant data; and the sixth and final section outlines the conclusions drawn from analysis of generated data, and presents opportunities for future work relevant to the field and to the results of the performed data generation and analysis.

## **1.3 Thesis Statement**

The Horizon Simulation Framework, as a readily adaptable, flexible, and versatile system modeling and simulation toolset, is an ideal modeling framework for use in the conceptual design and system requirements validation for the CubeSat Astronomy Network. The development of models using the Horizon Framework not only can inform and validate the design of the CubeSat Astronomy Network, but expand upon the current modeling methodologies available for modeling CubeSat missions. Thus, models for CubeSat systems shall be developed and utilized for validating the conceptual design requirements for day-in-the-life operations of the CubeSat Astronomy Network system, and shall be developed and validated such that future work on the CubeSat Astronomy Network shall be able to improve upon the existing CubeSat Astronomy Network models throughout its development cycle. The results of this work will be presented and published as the central focus of this thesis, principally as



a means assessing the feasibility of the conceptual design of the CubeSat Astronomy Network.

## Chapter 2

### BACKGROUND

#### 2.1 Survey of Undergraduate Astronomy

Existing literature on the involvement of students in the astronomical sciences suggests that the most effective means of educating new and future scientists entails the development of well-maintained and organized research teams. Research environments with students taking on apprentice-like roles, when engaged in peer mentoring and driven by compelling and achievable research objectives, have been observed to have the greatest success at retaining students in both communities of practice and wider networks of professionals [5]. Consequently, the inception and completion of astronomy research projects, especially for students in high school and undergraduate programs, is critical to the maintenance of these communities of practice and global networks. Fitzgerald et al. [6] identified nine critical criteria to classify existing astronomical research projects as relevant in-class learning experiences for high school students:

1. Capacity for original research
2. Student-obtained data from research-grade instruments
3. Focus on data interpretation
4. Reliance on standard astronomical methodologies and approaches
5. Focus on astronomical science
6. Active interaction between students and educators
7. Multi-teacher and/or multi-group involvement

8. Appropriateness for high school educators and students
9. Establishment and sustained operation

Similar criteria for success are applicable to educational research opportunities in astronomy for college undergraduate students. Reasonably-scoped research objectives in a limited-time project cycle have characterized previously successful and currently successful educational programs [7],[8]. Additionally, the acquisition of data from professionally designed and operated observation platforms and subsequent interpretation and presentation of original results has also been found to not only enhance student methodological and analytic skills; it also stimulates individual efforts by means of ensuring publication of their results in a fashion consistent with professional research methodologies and practices. This coupling of experiential learning in a true research environment with the intellectual demands and rigors of the data interpretation and publication process ensures that students gain methodological experience while also developing reasoning skill imperative to astronomical research.

While this holistic learning environment has been demonstrated to successfully train and retain new researchers in the field of astronomy, limited access to facilities and funding can inhibit research participation for students [6]. While the introduction of remote telescope systems has produced cost savings for the design and implementation of astronomical sensing systems, aggressive demand schedules for larger observatories can effectively limit the scope of student research due to proposal submission and scheduling latency. Therefore, alternative platforms, namely low-cost space-based telescopes, have been investigated for their suitability as research platforms to enable compelling research for undergraduate students. To begin the process of conceptual design for this proposed system, model-based systems engineering methodologies have been considered.

## 2.2 Model-Based Systems Engineering Methodologies

Several methodologies for model-based systems engineering have arisen and been surveyed in practice [1]. Many model-based systems engineering methodologies make use of SysML, a modeling language developed by the Open Modeling Group in 2007. SysML is a visual modelling language designed as an extension of the Unified Modeling Language (UML) [9], designed to overcome shortcomings of UML as a general system modeling paradigm. Many of these shortcomings were due to limitations in modelling requirements and parametric relationships. Modelling these relationships in SysML is less inhibited by the scope of its component elements [9]. Due to the visual nature of the models, individual models are created as diagrams with visual elements representing individual elements and relationships between elements. These diagram types can fall into one of four model categories: diagrams that model system structure, or the arrangement and interfaces of system elements and components; diagrams that model system behavior, including functions and interactions; diagrams that model requirements for the prior structures and behaviors; and parametric relationships between any element parameters or their behavior [9]. Previous work by Luther[10] and Patel[11] involved the development of functionality that enables models developed using SysML to be simulated in the Horizon Simulation Framework. These have some missing functionality due to the structure of the code utilized by the models, and the translator functions are unable to directly modify the underlying code as needed for the models to be properly simulated [10]. The addition of model scripting in the latest version of the Horizon Simulation Framework may prove sufficient to introduce this functionality [12], but it is not deemed an immediate necessity to introduce model generation via the translation of existing SysML models.

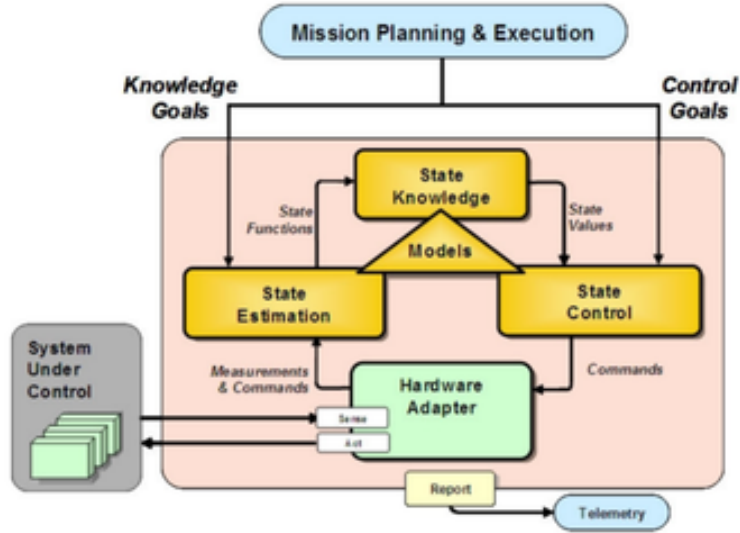


Figure 2.1: JPL State Analysis Control Architecture[1]

### 2.2.1 JPL State Analysis

While some methodologies are commercially-available, there are some published in open literature. One such example of a publicly-available methodology is the State Analysis methodology developed by JPL [1] and shown in Figure 2.1. One of the principle features of the State Analysis methodology is the use of states to represent instantaneous system conditions. These system states change over time according to the models developed to describe them. Important to note, however, is that the state referred to above is an extension of the classical control theory definition of state to include system component modes, available capacity for resources, and other physical or logical parameters that can be used to totally describe a systems instantaneous state of being. The primary activities of State Analysis include: state-based behavioral modelling, state-based software design, and goal-directed operations engineering. By means of these activities, system behavior is modelled by means of system state and how its component state variables relate to one another, validation and verification strategies for generated models are detailed and organized, and system mission objectives can be characterized and modelled by means of simulation

[1]. These activities can be accomplished and achieved by developing models utilizing the Horizon Simulation Framework, due to the design intent as a platform for use in conceptual design.

### **2.2.2 Extended Applications Beyond Conceptual Design**

Such a modeling methodology lends itself particularly effectively to characterizing fault analysis when system components and states create unfavorable system conditions. To prevent such unfavorable states from arising, modelers impose constraints on certain behaviors or parameters. This fundamentally changes the characterization of failure modes as arising from component failures to arising due to exceeded constraints on behavior or structure. As such, failure events are not directly modeled, but are prevented by designing a controller in the form of a constraint to prevent the failure from happening. [1] Subsystem failure analysis has already been studied using the Horizon Simulation Framework in previous work [13], and schedule failures are found using the built-in scheduling algorithms breadth-first search in order to identify successful schedules. Fault-tolerant designs, therefore, should include adequate constraints on system states and behaviors.

As models begin to expand over the course of project life-cycles, it may become necessary for external functionalities to be introduced to these modelling frameworks. To accomplish requirements verification, for example, may require the use of executable software to perform simulations and data analysis to verify that system requirements were satisfied. The Executable System Engineering Method developed at JPL [14] is an extension of the INCOSE OOSEM methodology [1], capable of using executable models to support requirements analysis and verification. Using these executable model results, documentation products associated with traditional systems engineering can also be supplied. This is similar in functionality to the Horizon Sim-

ulation Frameworks use of executables to run simulations and generate schedules of successful events.



## 2.3 CubeSat Model-Based Systems Engineering

To minimize total system cost for a space-based astronomy platform, the CubeSat specification has been primarily considered for development of space-based assets for the CubeSat Astronomy Network. [15]. MBSE methodologies for CubeSat missions and systems have been developed for numerous applications and multiple agencies and research teams.

INCOSE's Space Systems Working Group has been developing a CubeSat Reference Model using SysML [16]. The Radio Aurora Explorer CubeSat mission was selected as a reference mission the SSWG utilized in the course of developing the CubeSat modeling framework during the first two phases of development [17]. As of 2018, the CRM is in the process of being developed as an OMG specification [18], and has several packages implemented in the framework, including models of stakeholders and stakeholder needs, requirements models, use case and architecture packages, and technical measure packages for specifying system figures of merit. The CRM has also been implemented with No Magic's Cameo System Modeler, enabling models to be specified via Cameo System Modeler's graphical user interface. The CRM still has yet to be fully validated for model fidelity and function, and still requires implementation of parametric models for system mass, power, and cost at the CubeSat, subsystem, and component levels [18]. The development of robust parametric models for CubeSat systems will therefore be essential for valid model generation during the design of the CubeSat Astronomy Network. Previous work on the Horizon Simulation Framework implemented SysML model translation, successfully generating XML model files for SysML models [10], [11]. One of the modeling cases for validation was a SysML model of the ExoCube mission. However, the underlying Horizon subsystem models used

by the ExoCube model files were not modified to reflect the behavior and functions specified in the SysML models, meaning the translated models could not be correctly simulated for model validation. Subsystem model modification will be required to accurately model the CubeSat Astronomy Network, but no SysML models of the CubeSat Astronomy Network have been generated. Thus, SysML model translation validation is not necessary to generate and validate a system model of the CubeSat Astronomy Network, unless an existing SysML model were to be generated.

### **2.3.1 CubeSat MBSE Applications**

The CRM is just one of many methodologies capable of being utilized for model-based systems engineering of CubeSat systems. An MBSE framework was developed for development of the DelFFi mission [19], and utilized SysML for specification generation and requirement documentation. A parametric model was then simulated as an executable model in Simulink to validate the DelFFi system configuration and verify compliance with system requirements. This methodology is similar to what is utilized in the Horizon Simulation Framework, where requirements for subsystems can be directly specified as constraints to be satisfied during simulation. These modeling and simulation activities form a tripartite process for mission development:

1. Model Structure and Behavior Development
2. Model Architecture Development
3. Onboard Flight Software Development

Key to the MBSE methodology described [19] is the applicability of the developed models and architecture to the functional specification and development of required flight software itself. In essence, models generated for system validation can be used to form the logical and sometimes literal architecture of flight software.



Other mission developers have characterized the benefits of MBSE techniques for CubeSat mission development, such as Cipera’s analysis of the ease of iterating system designs upon development of baseline models [20]. Other applications of MBSE methodologies have included conceptual design [21], multi-satellite system optimization [22], [23] and program enterprise modeling [24]. These applications range from initial concept design to high-level programmatic modeling, demonstrating the range of potential applications of MBSE methodologies in CubeSat project life cycles. It is thus imperative that the Horizon Simulation Framework not only be useful as an MBSE methodology in the early stages of SSTN system design, but it must also be capable of further development with the project life cycle to provide improved model granularity and design validation.

## **2.4 CubeSat and Nanosatellite Astronomy Heritage and Development**

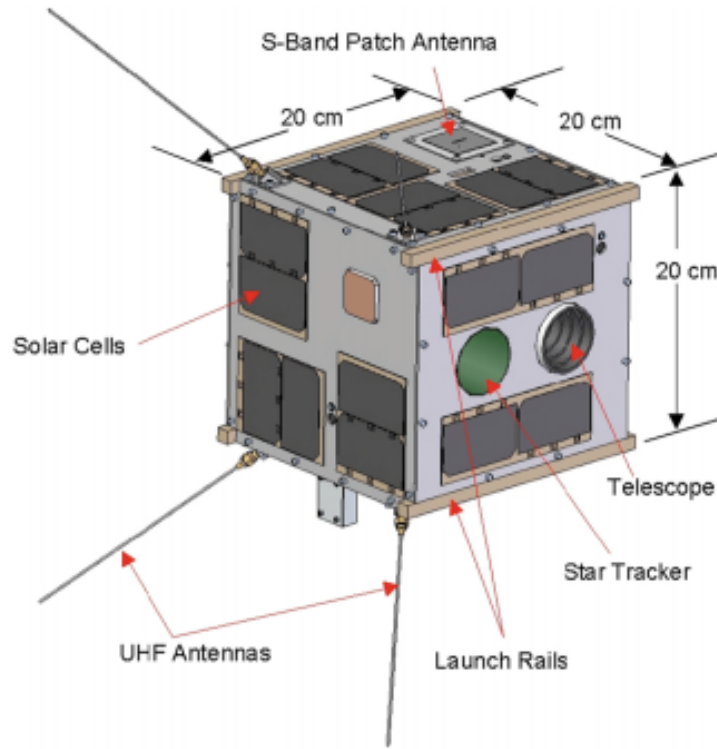
The limitations of science capabilities for nanosatellite stem primarily from instrument size constraints. Due to limitations on aperture size for smaller spacecraft, resolution and signal-to-noise capabilities are diminished by comparison with larger space-based and ground-based observatories. While design factors like deployable optics [25] and formation flight as a virtual telescope [26] can potentially augment existing small satellite astronomy capabilities, astronomical missions for nanosatellites and CubeSats presently remain limited in terms of individual instrument performance due to size constraints and other consequent design constraints. However, the advantages of nanosatellite systems compared to traditionally larger space telescopes include reduced per-unit non-recurring costs and the ability to launch and deploy multiple units as secondary payloads of launch vehicles, principally as a result of the lower launch mass per nanosatellite compared to larger spacecraft [27]. Thus, as was also noted by Cipera [20], there is the potential for alternative mission configurations, such as single spacecraft or multi-spacecraft systems, to achieve cost parity driven by the same re-

quirements with similar overall system performance, accounting for variances in power and communications requirements when leveraging different spacecraft mission architectures. Principally, the results of comparative analysis between a single-satellite earth-observing space system and a CubeSat-based constellation showed that though diminished effective ground resolution capability was demonstrated by the CubeSat constellation, roughly equivalent performance was achieved for data product latency (the time between an observed event and the downlinking of the recorded observation of said event) and system coverage (ground area observation capability during mission operations) between the single-satellite mission and the CubeSat constellation mission[20]. A multi-satellite formation or constellation mission architecture must therefore be considered in the course of system design. Such "system of systems" architectures for nanosatellite space systems have been demonstrated to be feasible and effective, enabling more ambitious concepts to be developed as nanosatellite development has become more accessible and cost-effective.

#### **2.4.1 Previous Nanosatellite Astronomy Missions**

Several nanosatellite-based missions have been demonstrated to be capable of performing valuable astrophysical data collection. The BRiGht Target Explorer mission (BRITE), an international collaborative mission between Austria, Canada, and Poland [28], consists of a constellation of five satellites, each satellite measuring 20 by 20 by 20 centimeters in edge length as a cubical bus as show in Figure 2.2 [2]. Each satellite is equipped with a 3 centimeter aperture, 24 degree field-of-view telescope for performing high-precision differential photometry on bright, low apparent magnitude stars. By operating as a constellation, individual fields of the night sky can be viewed simultaneously by all five satellites, or multiple fields can be imaged separately and concurrently. Observations of relatively luminous, high-temperature stars in the galactic plane are principal targets of interest for observation for BRITE,

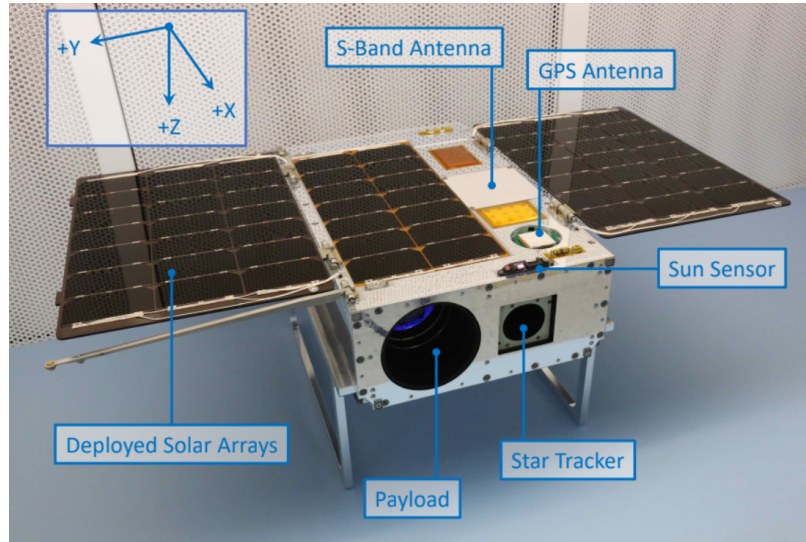
and research produced using data from BRITE include astroseismological studies of luminous stars, and the discovery of highly eccentric binary star systems. These results were able to be obtained from such small-aperture instruments due to the advantages of performing differential photometry from space, as opposed to utilizing a ground-based platform [29]. The scientific understanding provided by BRITE, obtained from a system of spacecraft roughly the size of an 8U CubeSat, demonstrate the compelling capabilities of nanosatellite astronomy.



**Figure 2.2: BRITE Spacecraft[2]**

More recently, the Arcsecond Space Telescope Enabling Research In Astrophysics (ASTERIA) [3] is an astronomical mission developed and operated by JPL to demonstrate the capabilities of detecting exoplanet transits using a 6U CubeSat platform, shown in Figure 2.3. ASTERIA's fine photometric precision required for detecting exoplanet transits is enabled by its payload design, maintaining a stable payload tem-

perature to within 0.01 Kelvin. A two-stage pointing control architecture, provided by a Blue Canyon Technologies XACT module for spacecraft attitude determination and control in addition to a two-axis piezoelectric-actuated imaging sensor, enable arcsecond-level pointing accuracy and sub-arcsecond stability for imaging operations. Precision photometry is enabled by this pointing system and allows ASTERIA to detect exoplanet transits of its three extended mission science targets: 55 Cancri, HD 219134, and Alpha Centauri [30]. Prior work in modeling aerospace attitude control systems has been accomplished using the Horizon Simulation Framework [31], though a model of sufficient fidelity for validation through simulation has not yet been successfully developed for a CubeSat spacecraft model.



**Figure 2.3: ASTERIA 6U CubeSat Spacecraft[3]**

#### **2.4.2 Current Development of Nanosatellite Astronomy Missions**

Recent literature by Shkolnik [32] and others anticipate a growth in the use of CubeSat systems in astronomical science missions. Due to their relatively lower development costs compared to larger astronomical space systems, CubeSat systems are well-suited for complementing the capabilities of costlier dedicated science platforms,

especially time domain observations and, when utilized in constellations, enhanced target field coverage or simultaneous coverage of multiple fields. One of the major science capability gaps that has been identified to be suitable for CubeSat astronomy missions is in the performance of time domain astronomy in photometric bands normally inaccessible to ground based observatories. Some electromagnetic wavelengths are normally poorly transmitted through Earth’s atmosphere, but can be detected from space in the absence of an interfering atmosphere. Proposed CubeSat missions to demonstrate astronomy capabilities at wavelengths normally undetectable by ground-based observatories include the Colorado Ultraviolet Transit Experiment and the Star-Planet Activity Research CubeSat[32]. These missions are being designed to observe exoplanets in near and far ultraviolet wavelengths to complement the capabilities of other astronomical systems. Additionally, improvements in CubeSat lifetime and reliability, and recent demonstrations of optical communications and attitude control systems, have made CubeSats even more viable for astronomy instrument platforms. All these factors considered make CubeSat systems an increasingly attractive option for broad adoption and development; thus, the design and testing of robust modeling and simulation frameworks for CubeSat astronomy systems is similarly becoming more desirable and necessary.

## **2.5 CubeSat Astronomy Network**

The CubeSat Astronomy Network is a proposed mission to perform astronomical surveys of approved targets proposed by research teams of undergraduates. A previously-conducted conceptual design study performed in 2018 produced the mission concept as further described below[4]. By leveraging relatively less costly CubeSat-based platforms, low-cost access to space-based astronomical data products can be leveraged for the purpose of fostering novel research opportunities for undergraduate research teams. As demonstrated by the BRITE [28] and ASTERIA [30] missions,

nanosatellite-based astronomy missions can enable research in high-priority science objectives, such as exoplanet discovery and characterization. Given the relatively low cost and potential of CubeSat-based systems for facilitating astronomical research, a novel 12U design has been investigated for operation as part of a constellation of CubeSat space telescopes[4].

### **2.5.1 System Architecture**

From the system-level, the CubeSat Astronomy Network can be considered to be a distributed system-of-systems composed of two segments: a Ground Segment and a Space Segment. While these segments can be further divided into lower-level assets and elements, this distinction is relevant for how the system is modeled in the Horizon Simulation Framework.

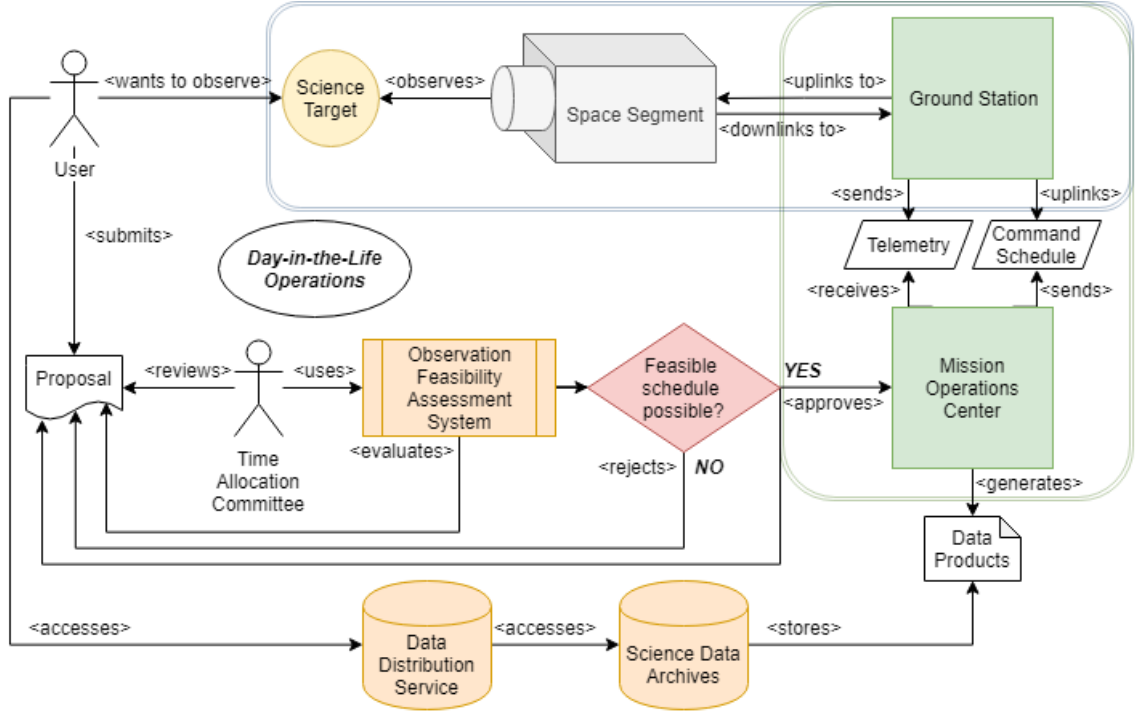
The Space Segment is composed of individual CubeSat Space Telescopes that execute commands either autonomously or remotely when commanded via the Ground Segment. Operations performed by the telescopes include: slew maneuvers for pointing the imaging payload towards target fields for imaging operations, or pointing a directional communication antenna for command uplink and telemetry downlink, or pointing as part of a contingency mode; power generation, storage, and distribution to subsystems and components; payload imaging operations, calibration, and status monitoring; image compression and processing; command and data handling; desaturation of reaction wheels as necessary to ensure pointing capabilities are maintained; subsystem health checks; and others not mentioned here. Each individual telescope, or "asset", will be capable of executing these operations individually according to an automatic task scheduler as an element of the flight software.

The Ground Segment consists of all system elements necessary to the nominal function of the CubeSat Astronomy Network that are not allocated to the Space Segment. This includes the processes of proposing and approving astronomical obser-

vation targets, observation scheduling, and distribution of data products. A Mission Operations Center will house the necessary hardware and equipment for system monitoring and command scheduling as necessary. Direct proximity to Space Segment Communication hardware is not necessary, but a line of communication between this center and the Space Segment will be necessary to ensure that observation schedules, once generated, are relayed to the assets in the Space Segment. Additionally, a line of communication between the Space Segment and a means of distributing approved data products, once downlinked and processed from the Space Segment, will allow users to access the science products pertinent to the targets they had proposed and were subsequently accepted for observation[4]. Users are not expected to supply direct commands to the Space Segment: rather, they are to submit proposals for observation targets and relevant observation parameters, which shall be reviewed by a panel of subject experts. If the observation targets and parameters, having been translated into tasks that can be executed by the Space Segment, can be successfully integrated and scheduled into system operations, then an observation schedule including these proposed targets shall be generated and uplinked to the Space Segment for autonomous execution, as more fully described in the concept of operations.

### **2.5.2 Day-in-the-Life Concept of Operations**

As per Figure 2.4, the high-level system concept of day-in-the-life operations for the CubeSat Astronomy Network is similar to how existing networked robotic observatories like the Las Cumbres Observatory [33] perform observations of science targets, from acceptance of proposed targets to delivery of science data products. The principal distinction between existing robotic telescope systems and the CubeSat Astronomy Network lies in the utilization of a Space Segment in the CubeSat Astronomy Network, as opposed to a network of robotic ground-based telescopes. Proposed targets, to be accepted and integrated into an observation schedule by the



**Figure 2.4: CubeSat Astronomy Network Concept of Operations**

Time Allocation Committee, must not be constrained such that their observation parameters, such as start and end times, conflict with the observational requirements of other targets, or cause the performing assets to exceed their operational capabilities. For a single CubeSat telescope, these inviolable constraints may include a maximum accumulated momentum in reaction wheels, or a pointing constraint due to the relative positions of the Earth, Sun, and Moon at a particular proposed observation time. Formal methods of describing such resource allocation schemes have been developed with these automated scheduling methodologies in mind [34].

Subsequently, once an observation has been approved and a command schedule has been generated, allocating tasks to individual assets as appropriate to ensure all observation requirements are satisfied, an uplink window is awaited for individual satellites by the Ground Segment communication facilities. Once link is established between a telescope and the Ground Segment facility, the command schedule is transmitted to the telescope. Upon receipt of the command schedule and cessation



of communication activities during the window, the telescope will begin the execution of commands autonomously to perform such observations and other necessary on-orbit activities to maintain system functionality, such as desaturating reaction wheels, charging batteries, and other operations[4].

### **2.5.3 Key and Driving Requirements**

The primary science capability of interest for the present iteration of the CubeSat Astronomy Network is the ability to perform time-series photometry on targets of interest [4]. Science targets of interest may include supernovae, exoplanet candidates identified by previous exoplanet surveys, and other science targets of interest for other CubeSat astronomical research missions [32]. Since exoplanet transits may occur over the duration of several hours, the capability for extended observation periods would deliver unprecedented capability in time-series photometry for a CubeSat space telescope. The performance of such observations from a spacecraft in low Earth orbit may be constrained by several factors, such as the performance of mission activities by a telescope that directly conflict with the ability to perform such observations, or unfavorable observation conditions due to the orientation of the Earth, Sun, Moon, and other celestial bodies during a particular time period. Other constraints may include pointing constraints that may restrict the areas of the sky that the telescope may be able to observe due to the adverse effects of atomic oxygen in a low Earth orbit that would adversely affect an imaging payload’s optical elements if they were pointed in the ram direction, as described for a variety of material types by Reddy [35]. Consequently, a summary of key and driving system-level requirements for the CubeSat Astronomy Network to enable continuous time-series photometry from a low Earth orbit are summarized below for its present iteration:

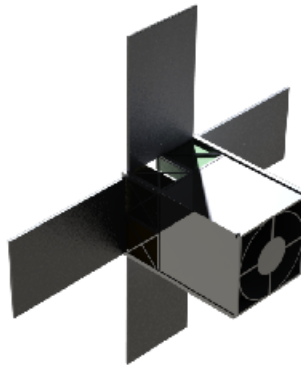
- The system shall perform time series optical observations of individual stellar

targets for no less than 90 percent of a target's specified observation cadence cycle.

- The system shall be composed of two primary operational segments: a Ground Segment and a Space Segment.
- The system shall include multiple space telescopes operating in a low Earth sun synchronous orbit.
- The system space elements shall orbit the Earth with a dusk-dawn local crossing time.
- The system shall be capable of imaging targets for exposure times longer than 60 seconds.
- The system shall directly downlink science data to a receiving ground station.
- The system shall reserve no less than 20 percent of its onboard mission data memory to remain available.
- The system shall discharge no more than 15 percent of its onboard total electrical charge capacity.
- The system shall point its observation instrument no further than 15 degrees below the local horizon plane in its orbit.
- The system shall point its solar panels no further than 45 degrees from the direction of the Sun.
- The system shall point its observation instrument bore axis no less than 75 degrees into the local ram direction in its orbit.

#### 2.5.4 Bus Design

Prior analysis concluded that a 12U CubeSat bus format would potentially be capable of satisfying these requirements. The larger internal volume than contemporary 6U-based astronomy missions would enable the integration of a larger aperture telescope payload than had previously flown in a CubeSat mission with a dedicated astronomical research role[4]. To that end, a conceptual design study[4] was conducted to characterize the key characteristics for such a mission concept, investigating a variety of relevant science targets and hardware options. A render of this conceptual design is shown in the Figure 2.5.



**Figure 2.5: Telescope Conceptual Design[4]**

Key design parameters for the conceptual design components and subsystems are shown in Table 2.1.

## 2.6 Horizon Simulation Framework

The Horizon Simulation Framework is a modeling and simulation framework originally developed to model and simulate the performance of aerospace systems. It has been developed through several versions, and its present version is written in C# and

IronPython, an implementation of Python that can leverage the features of the .NET Framework [12]. The principal application of the Horizon Simulation Framework over the course of its development has been for modeling and simulating the performance of aerospace systems by simulating the execution of required system tasks by systems. A collection of successfully executed tasks that satisfy user-specified constraints is referred to as a schedule. These systems are composed of subsystems whose behavior and parameters are designed by a model developer to model the desired system activities and characteristics.

### **2.6.1 Glossary of Horizon Terminology**

Due to the specificity of terms used when describing the Horizon Simulation Framework, a common set of definitions has been used in previous work to describe concepts and elements of the framework [37] [12].

- **Asset:** An individual element of a single or multi-element System that contains Subsystems and executes Tasks. Used to describe a collection of Subsystems that share the same Dynamic State.
- **Constraint:** An acceptable value for a State. A logical check is performed by the scheduling algorithm during the simulation process to ensure that all States with Constraints, upon the execution of a Task at some time, satisfy the conditions described by those Constraints.
- **Dependency:** A relationship between Subsystems in a System. When specified, a Dependency function can be used by one Subsystem to communicate relevant State information to another System, allowing System functionality and States to be distributed amongst Subsystems instead of being globally scoped.
- **Dynamic State:** A designated State for describing dynamic information about an Asset, such as position or velocity.

- Event: The execution of a Task at a simulation time. At every simulation time step an Event occurs and a Task can be executed between the start and end times of an Event.
- Schedule: A collection of Events executed by the system. Schedules of proposed sequences of Events are generated by the scheduling algorithm at each time step, and schedules whose Constraints are satisfied are retained for generating Schedules of Events that can occur at future timesteps.
- Subsystem: A specification of related attributes and functions of a subset of a System Asset. Determines whether Tasks can be successfully executed by the System and update their State if a Task can be successfully executed.
- System: The collection of Assets that are modeled in the Horizon Simulation Framework.
- Task: A required type of System activity for an Asset to perform when a Target is available.
- Target: A required goal or objective for which a Task is performed. Examples include ground facilities for communication with an Asset or a location on Earth's surface to be imaged.

### **2.6.2 Architecture**

An important feature of Horizon is the decoupling of aspects of simulation and aspects of modeling. Models developed using Horizon specify the behavior and parametric relationships between subsystem states, tasks executed by the systems, and the corresponding target assets for tasks. The scheduling of tasks performed by the system, however, is handled by separate code than what is used to model a system. This distinction between scheduling tasks and simulating system behavior and describing the

underlying structure and parameters that model that behavior ensures that users can model systems without the need to create a unique task scheduling and evaluation system for every system model. In essence, the scheduling algorithm checks that tasks are performed in proposed schedules generated by the algorithm, and utilizes functionality specified by the subsystem models to evaluate schedules for conformance.

### **2.6.3 Scheduling Algorithm**

To simulate the successful execution of tasks by the system, a scheduling algorithm is used to generate schedules of events executed over the simulation time domain specified by the user. Possible events at a given simulation time step are evaluated by each asset's subsystem models to see if the proposed task can be successfully performed by each subsystem while satisfying constraints. Subsystems whose states must satisfy constraints during an event but whose behavior is not controlled by a state machine within the subsystem model are also checked to ensure that their constraints are satisfied over the entire period of time over which an event occurs. These are principally modeled using a subsystem's `canPerform()` or `canExtend()` methods, whereby the ability to perform a task and update its state is checked by the subsystem using the prior, or its ability to propagate its state such that all constraints are satisfied during an event using the latter. To evaluate whether dependencies are satisfied, a subsystem model's dependency functions are used to send relevant state information to the dependant subsystem, whereby the dependant subsystem's state can thus be evaluated. This scheduling algorithm performs a breadth-first search of possible schedules and events to evaluate which schedules can satisfy subsystem constraints. It does so by evaluating schedules and assigning them a value: this schedule evaluator can be specified by a user to weight generated schedules as more or less desirable. A finite number of schedules can then be stored and propagated in the time domain such that the most valuable schedules at a simulation time step are propagated, while

less valuable schedules are discarded. This is similar to the methodology of "branch-and-bound" strategies in dynamic programming, as described by Moghaddam and Usher [38]. This can be leveraged for a system designer to prioritize the generation of schedules that perform the most desirable tasks and the least desirable ones according.

While a user can theoretically generate every possible successful schedules over the course of the time domain specified as an input to the system, it may be more useful for a user to generate only a limited number of schedules at every time step. This can be accomplished by selecting a maximum number of schedules to retain in memory at a time. Once this number of potential schedules has been generated at a particular time step, they are sorted from most to least valuable. If a new schedule is generated of higher value, then the current lowest value schedule is removed from this list of schedules, and the simulation continues to propagate. If a user crops only to one schedule per time step, then the scheduling algorithm behaves like a greedy search algorithm, selecting the locally highest-value potential schedule at every time step. While this may be acceptable for modeling deterministic systems, this can prove troublesome if a series of locally-optimal scheduling steps result in a system state that ultimately fails to meet requirements. Therefore, it may be more prudent to crop to a relatively modest number of potential schedules to prevent such undesirable simulation results. While it is possible to retain every potential schedule generated to determine, within the specified conditions of the simulation, the globally optimal schedule of events for a system, users may be constrained by their access to processing power and memory, especially for systems with many degrees of freedom or multiple assets executing tasks.

The time complexity of the scheduling algorithm implemented in the Horizon Simulation Framework was characterized by means of a parametric time study for an earlier version of the framework before its present incarnation. The results of this analysis, performed by O'Connor [39] determined that the existing exhaustive

search algorithm had a time complexity of  $O(T^A S_{max} n_t t_{sys})$ , where  $T$  is the number of tasks and targets for each asset,  $A$  is the number of assets in the simulation,  $n_t$  is the number of individual scheduler time steps, and  $t_{sys}$  is the mean system execution time. No parametric study on number of subsystems or time complexity of subsystem methods has been conducted for the Horizon Simulation Framework, though such analysis may be warranted for future development of the Framework: however, such characterization of system performance falls beyond the scope of this thesis.

#### 2.6.4 Modeling Segment

A distinction between modeling and scheduling is central to the functionality of the Horizon Simulation Framework. Due to the structure of the underlying code, the scheduling algorithm itself is capable of generating potential schedules based on the state machines modeled within each subsystem model, and the results of these subsystems' `canPerform()` and `canExtend()` methods are used to assess the feasibility of accomplishing a given task for a proposed event. Since these methods return boolean values, and the methods used to check subsystem constraints and dependencies similarly return boolean values, the scheduling algorithm does not require knowledge of the particular state information or functionality specified within the model. This permits a degree of flexibility in the types of models that a user can implement in Horizon. However, circular dependencies between subsystems are not permitted in a system model, because evaluating the satisfaction of dependent relationships would require information from another subsystem or series of subsystems, it or themselves being dependent on the subsystem to be evaluated, such that the information required by the scheduler to schedule the event requires itself to already exist in order to be calculated. Logically, these circular dependencies cannot be resolved, so a "waterfall"



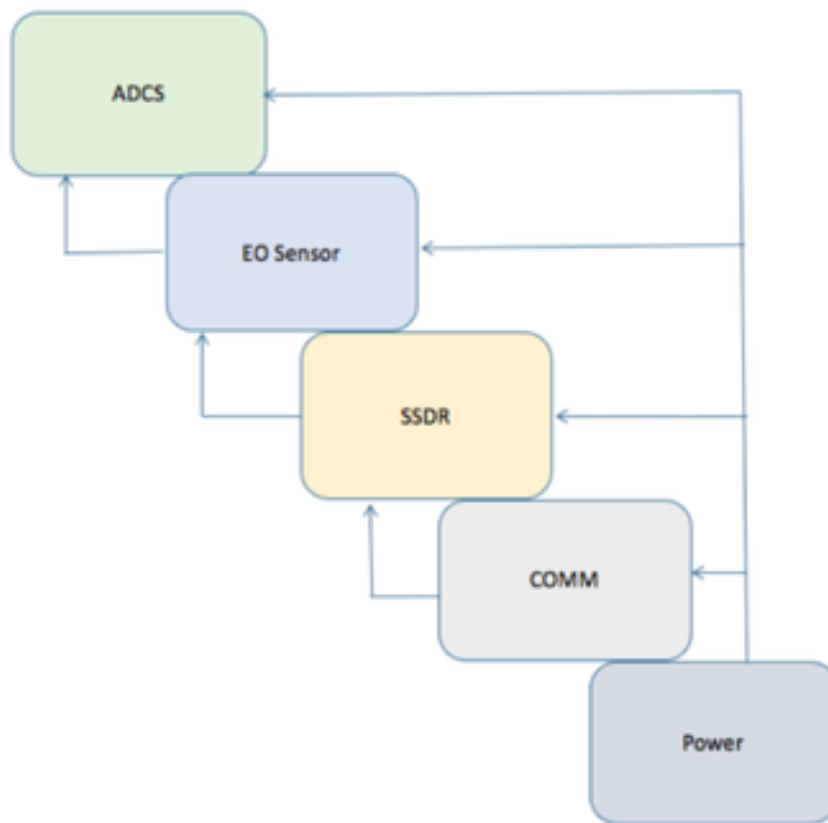


Figure 2.6: Example Horizon Dependency Diagram

dependency architecture is advised for specifying subsystem dependencies in system models. An example of such a dependency hierarchy is shown in Figure 2.6 [12].

### **2.6.5 Applicable Prior Work**

The 3.0 version of the Horizon Simulation Framework was developed in C# with Python scripting functionality by Yost [12], with additional text-based logging functionality introduced by Lunsford [13] in 2016. The results of bringing the Horizon Simulation Framework into version 3.0 from version 2.3 had the effects of improving run time performance and simplifying the underlying architecture of Horizon's code. Integration with the .NET Framework permits a wide access to namespaces, as well as the ability to introduce additional software functionality through the use of NuGet packages. These packages are used to provide access to IronPython, an open-source implementation of Python that is implemented with the .NET Framework. Scripted models written in IronPython that are accessed by Horizon are written syntactically identically to Python, but can import namespaces and packages accessible when using the .NET Framework. Furthermore, data types specified in the Horizon Simulation Framework can be utilized in these scripted models, permitting the re-use of C# types in a language that is otherwise indistinguishable from Python. This is particularly advantageous for new users that have a background in Python, as even relatively inexperienced Python users can learn relatively quickly. An understanding of the underlying C# models is still necessary for a user to be able to effectively model a system, as the Horizon-specified data types may leverage types differently than standard types in Python, as characterized by Yost [12].

Research by Maclean [31] and Frye [40] demonstrated the applicability of leveraging Horizon models for simulating the performance of guidance, navigation, and control systems. Maclean's research involved the development and flight test validation of a modeled flight control architecture for a sounding rocket, and demonstrated

the application of Horizon for modeling linear-quadratic regulator (LQR) flight controllers and state estimation. Frye’s research modeled the performance of a swarm of reconnaissance UAV’s through the use of digital pheromones and pheromone mapping. Like Maclean’s model, Frye’s UAV assets leveraged LQR controllers for maintaining flight stability. Both Maclean’s and Frye’s models were principally developed using IronPython models accessed by the Horizon code written in C#, demonstrating the applicability of scripted models in the Horizon Simulation Framework.

**Table 2.1: Telescope Conceptual Design Key Parameters**

Parameter	Value
Science Mission Concept	Long-Duration Time Series Photometry
Bus Format	12U CubeSat (20 by 20 by 30 cm)
Payload	Approx. 18 cm Aperture 2-Mirror Telescope Imaging Photometer
Pointing Capability	Three-Axis Slewing Momentum Bias Stabilization
ADCS Module	Blue Canyon XACT
ADCS Actuators	RWP-015 Reaction Wheels (included w/ XACT)
ADCS Desaturation Strategy	Mag. Torque Rods XACT and ISIS iMTQ Board [36]
Pointing Modes	Boresight-Constrained Target Pointing Local Horizon Alignment Anti-Sun Alignment
Downlink Method	S-Band Patch Antenna
Power Generation	Body-Fixed One-Time Deployable Solar Arrays

## Chapter 3

### METHODOLOGY

For the purposes of characterizing the Concept of Operations of the CubeSat Astronomy Network, and the advantages of constellation-based astronomy missions compared to monolithic single-satellite missions, a pair of design reference missions were developed. Each mission consists of either a single or multiple CubeSat Space Telescopes and a Ground Segment consisting of a communications facility at California Polytechnic State University’s ground station located on its main campus in San Luis Obispo [41]. Individual telescopes are modeled as Assets in a Horizon Model XML file, each telescope composed of an Attitude Determination and Control (ADCS) subsystem, a Payload subsystem, a Mission Data Handling (MDH) subsystem, a Communications subsystem, and a Power subsystem. Additionally, each spacecraft has a unique Dynamic State describing its position, velocity, attitude, and body angular velocity in the Earth-centered inertial frame. These models are adapted from the existing Aeolus reference mission models described by O’Connor et al. [37] and Yost [12], though the existing models in their configurations described by O’Connor and Yost were deemed insufficient for characterizing the CubeSat Astronomy Network reference mission at this stage of development. The subsystem and asset dynamics models are described in greater detail in section 3.2 and section 3.3.

#### **3.1 Reference Frames**

For the purposes of modeling the CubeSat Astronomy Network, there are three key reference frames: the Earth-Centered Inertial reference frame, the Local Horizontal reference frame, and the Body frame.

### 3.1.1 Earth-Centered Inertial Frame

The Earth-Centered Inertial Frame is defined per the below equation using vectrix notation [42], where its origin is at the Earth's center,  $\hat{\mathbf{x}}_0$  points in the direction of the vernal equinox,  $\hat{\mathbf{z}}_0$  points towards celestial north, and  $\hat{\mathbf{y}}_0$  is orthonormal to  $\hat{\mathbf{x}}_0$  and  $\hat{\mathbf{z}}_0$  such that  $\hat{\mathbf{y}}_0 = \hat{\mathbf{z}}_0 \times \hat{\mathbf{x}}_0$  [42]:

$$\vec{\mathcal{F}}_0 = \begin{bmatrix} \hat{\mathbf{x}}_0 & \hat{\mathbf{y}}_0 & \hat{\mathbf{z}}_0 \end{bmatrix}^T \quad (3.1)$$

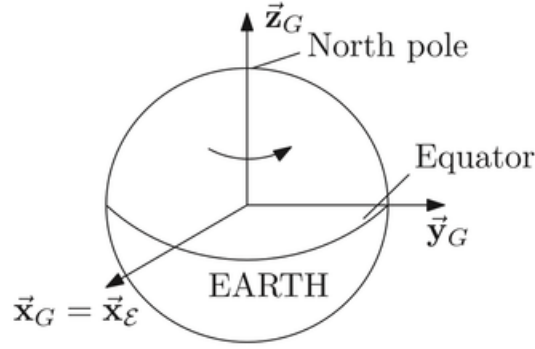


Figure 3.1: ECI Reference Frame

### 3.1.2 Local Horizontal Frame

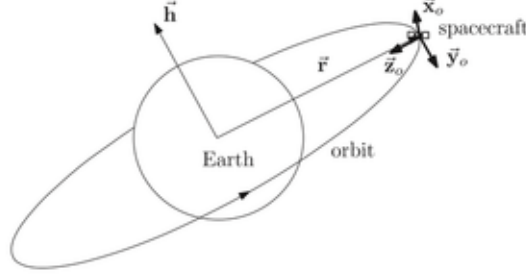
A local orbiting reference frame [42] that varies with a spacecraft's position and velocity in an orbit defined in the inertial frame, referred to as a local horizontal frame, can be defined using an orbital position  $\mathbf{r}_0$  and an orbital velocity  $\mathbf{v}_0$  with respect to the inertial frame as:

$$\vec{\mathcal{F}}_\lambda = \begin{bmatrix} \hat{\mathbf{x}}_\lambda & \hat{\mathbf{y}}_\lambda & \hat{\mathbf{z}}_\lambda \end{bmatrix}^T \quad (3.2)$$

where the axes are defined as follows, with  $\hat{\mathbf{z}}_\lambda$  pointed towards nadir as shown in Figure 3.2:

$$\hat{\mathbf{x}}_\lambda = \hat{\mathbf{y}}_\lambda \times \hat{\mathbf{z}}_\lambda, \quad \hat{\mathbf{y}}_\lambda = -\frac{\mathbf{r}_0 \times \mathbf{v}_0}{\|\mathbf{r}_0 \times \mathbf{v}_0\|}, \quad \hat{\mathbf{z}}_\lambda = -\frac{\mathbf{r}_0}{\|\mathbf{r}_0\|} \quad (3.3)$$

where  $\mathbf{r}_0$  is the orbital position in the inertial frame and  $\mathbf{v}_0$  is the orbital velocity in the inertial frame.



**Figure 3.2: Local Horizontal Reference Frame**

The direction cosine matrix that describes the rotation from inertial frame coordinates to local horizontal frame components is defined as:

$$\mathbf{C}_{\lambda,0} = \vec{\mathcal{F}}_{\lambda} \cdot \vec{\mathcal{F}}_0^T \quad (3.4)$$

$$\vec{\mathcal{F}}_{\lambda} \cdot \vec{\mathcal{F}}_0^T = \begin{bmatrix} \hat{\mathbf{x}}_{\lambda} \\ \hat{\mathbf{y}}_{\lambda} \\ \hat{\mathbf{z}}_{\lambda} \end{bmatrix} \cdot \begin{bmatrix} \hat{\mathbf{x}}_0 & \hat{\mathbf{y}}_0 & \hat{\mathbf{z}}_0 \end{bmatrix} = \begin{bmatrix} \hat{\mathbf{x}}_{\lambda} \cdot \hat{\mathbf{x}}_0 & \hat{\mathbf{x}}_{\lambda} \cdot \hat{\mathbf{y}}_0 & \hat{\mathbf{x}}_{\lambda} \cdot \hat{\mathbf{z}}_0 \\ \hat{\mathbf{y}}_{\lambda} \cdot \hat{\mathbf{x}}_0 & \hat{\mathbf{y}}_{\lambda} \cdot \hat{\mathbf{y}}_0 & \hat{\mathbf{y}}_{\lambda} \cdot \hat{\mathbf{z}}_0 \\ \hat{\mathbf{z}}_{\lambda} \cdot \hat{\mathbf{x}}_0 & \hat{\mathbf{z}}_{\lambda} \cdot \hat{\mathbf{y}}_0 & \hat{\mathbf{z}}_{\lambda} \cdot \hat{\mathbf{z}}_0 \end{bmatrix} \quad (3.5)$$

A quaternion can also be used to represent this rotation. Quaternions used as state variables in attitude controllers, as opposed to rotation matrices or Euler angles, can prove more efficient and less memory due to their use of linear algebraic operations to represent successive rotations. Euler angles, in contrast, require the use of trigonometric operators to be propagated forward in time for use in a full-state feedback controller, which can require longer processing times per propagation in a real-time controller. Rotation matrices do not require similarly complex operations, but in practice require the use of nine numeric elements to represent a single rotation, while quaternions require the use of only four to represent a single rotation.

Hence, to improve simulation runtime performance and computational efficiency, attitude kinematics in the Horizon Simulation Framework are by default modeled using quaternions, and their first use in modeling flight dynamics in HSF 3.0 was demonstrated by Maclean [31]. Quaternions defined using the scalar first specification are defined as:

$$\mathbf{q} = q_0 + q_1\hat{i} + q_2\hat{j} + q_3\hat{k} = \begin{bmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \end{bmatrix} = \begin{bmatrix} \eta \\ \epsilon_1 \\ \epsilon_2 \\ \epsilon_3 \end{bmatrix} = \begin{bmatrix} \eta \\ \boldsymbol{\epsilon} \end{bmatrix} \quad (3.6)$$

The quaternion used to describe the rotation from inertial to local horizontal coordinates is defined as:

$$\mathbf{q}_{\lambda 0} = \begin{bmatrix} \eta_{\lambda 0} \\ \boldsymbol{\epsilon}_{\lambda 0} \end{bmatrix} = \begin{bmatrix} \frac{1}{2}\sqrt{\text{trace}(\mathbf{C}_{\lambda 0}) + 1} \\ \frac{C_{23}-C_{32}}{4\eta_{\lambda 0}} \\ \frac{C_{31}-C_{13}}{4\eta_{\lambda 0}} \\ \frac{C_{12}-C_{21}}{4\eta_{\lambda 0}} \end{bmatrix} \quad (3.7)$$

where  $C_{ij}$  denotes the matrix element of  $\mathbf{C}_{\lambda 0}$  in the i-th row and the j-th column.

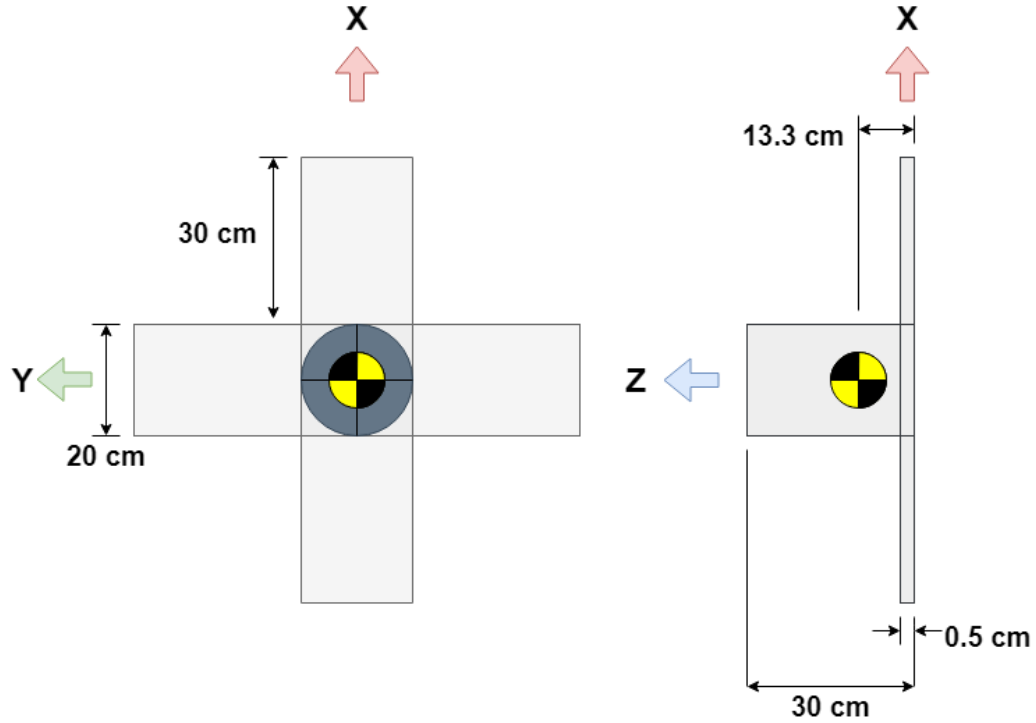
### 3.1.3 Body Frame

The body frame for the telescope body is defined per the below vectrix:

$$\vec{\mathcal{F}}_b = \begin{bmatrix} \hat{\mathbf{x}}_b & \hat{\mathbf{y}}_b & \hat{\mathbf{z}}_b \end{bmatrix}^T \quad (3.8)$$

Body axes for the telescope body are oriented per existing literature on 12U specifications [43] at the center of mass, with  $\hat{\mathbf{z}}_b$  aligned with the outwards payload instrument bore axis,  $\hat{\mathbf{x}}_b$  oriented opposite the side to which the antenna is mounted, and  $\hat{\mathbf{y}}_b$  completing the axes such that  $\hat{\mathbf{y}}_b = \hat{\mathbf{z}}_b \times \hat{\mathbf{x}}_b$ . This is illustrated in Figure 3.3.





**Figure 3.3: Telescope Body Axes and Approximate Center of Mass**

### 3.2 Model Subsystem Architecture

Three types of system tasks are modeled for the CubeSat Astronomy Network reference mission: imaging tasks for science targets, downlink tasks for transmitting science data, and desaturation tasks for desaturating reaction wheels, as shown in Table 3.1.

**Table 3.1: Table of System Tasks and Targets**

Task Type	Target Type
Image astronomy target	LocationTarget
Downlink astronomy image data	FacilityTarget
Desaturate reaction wheels	Desaturate

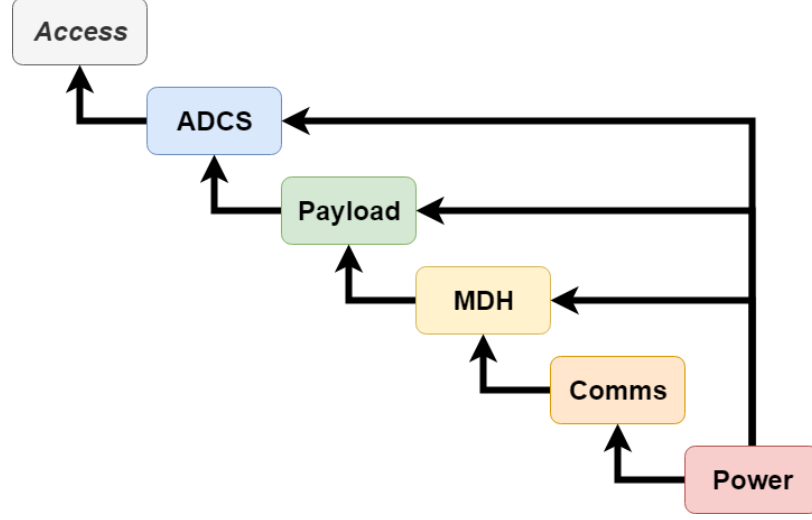
These tasks are checked by the scheduling algorithm during a proposed event for

an asset by each subsystem's `canPerform()` and `canExtend()` methods. Subsystems which are dependent on other subsystems for state information in order to evaluate their ability to perform a task have dependency functions to send relevant state information. Dependent subsystems can then access state information from other subsystems by means of these dependency functions. A diagram of the dependency relationships for each asset is shown in Figure 3.4. While the dependency relationships are fundamentally identical to those of the Aeolus model dependencies, several constraints, parameters, and dependency functions, in addition to entirely new functionality, have been added to these models.

Note that a subsystem named "Access" has been included in this diagram. The Access subsystem in Horizon is not a subsystem that necessarily has a strictly-related hardware or software analogue in the CubeSat Astronomy Network. It is a subsystem for which the ADCS subsystem has a dependency, and has been used in this model architecture to provide the ADCS subsystem with the times when targets are visible to an asset when not blocked by the Earth. The times during these access windows when line-of-sight visibility is achievable between the asset and the target are used by the scheduling algorithm to propose event time spans during which tasks can be executed. This functionality was utilized in the Aeolus test case simulations to ensure that ground targets were visible by an orbiting asset at the time of a proposed event by the scheduling algorithm. It fulfills a similar purpose in modeling and simulating the CubeSat Astronomy Network, and has thus been included for completeness due to its specification in the model files used during the development of the models used for this thesis.

### 3.3 Dynamics Models

Prior development activities for the CubeSat Astronomy Network included the estimation of mass properties of the spacecraft bus to estimate orbital lifetime[4].



**Figure 3.4: Asset Subsystem Dependencies for the CubeSat Astronomy Network**

A table of the spacecraft mass properties used to model the system in the Horizon Simulation Framework is shown in Table 3.2.

The dynamic state of a single CubeSat Space Telescope is described using the following equations of motion. The state variables that compose the total dynamic state description of a CubeSat Space Telescope are as follows:

1. Position in the ECI frame  $[\mathbf{r}_o]$
2. Velocity in the ECI frame  $[\mathbf{v}_o]$
3. Quaternion describing the orientation of the body frame with respect to the ECI frame  $[\mathbf{q}_{bo}]$
4. Angular rates of the body frame in the ECI frame, with respect to the body frame  $[\boldsymbol{\omega}_{bo}]$
5. Reaction wheel angular speeds, defined with respect to the body frame  $[\boldsymbol{\omega}_{wb}]$

**Table 3.2: Conceptual Design Mass Properties**

Parameter	Value
Mass	8.5 kg
$J_{xx}$	$0.15 \text{ kg} - m^2$
$J_{yy}$	$0.15 \text{ kg} - m^2$
$J_{zz}$	$0.15 \text{ kg} - m^2$
$J_{xy} \approx J_{xz} \approx J_{yz}$	$0 \text{ kg} - m^2$
<b>J</b>	$\begin{bmatrix} J_{xx} & 0 & 0 \\ 0 & J_{yy} & 0 \\ 0 & 0 & J_{zz} \end{bmatrix}$

As represented in the below state vector:

$$\mathbf{x} = \begin{bmatrix} \mathbf{r}_0 \\ \mathbf{v}_0 \\ \mathbf{q}_{b0} \\ \boldsymbol{\omega}_{b0} \\ \boldsymbol{\omega}_{wb} \end{bmatrix} \quad (3.9)$$

These states are numerically integrated over the simulation time domain in Horizon using a Runge-Kutta scheme. By default, a Dormand-Prince integration method is used for state propagation in the Horizon Simulation Framework.

Orbital position is propagated per the below differential equation:

$$\dot{\mathbf{r}}_0 = \begin{bmatrix} \dot{x}_0 \\ \dot{y}_0 \\ \dot{z}_0 \end{bmatrix} = \mathbf{v}_0 \quad (3.10)$$

Orbital velocity is propagated per the below differential equation:

$$\dot{\mathbf{v}}_0 = \mathbf{a}_{2-body} + \mathbf{a}_{J2} + \mathbf{a}_{drag} \quad (3.11)$$

Two-body orbital acceleration is calculated according to the below nonlinear differential equation.

$$\mathbf{a}_{2-body} = -\frac{\mu}{\|\mathbf{r}_o\|^3} \mathbf{r}_o \quad (3.12)$$

Orbital perturbation acceleration due to Earth oblateness is modeled according to the below equation per De Ruiter et al. [42].

$$\mathbf{a}_{J_2} = \frac{3\mu J_2 R_e^2}{2 \|\mathbf{r}_o\|^5} \left[ \left( 5 \frac{z_0}{\|\mathbf{r}_o\|^2} - 1 \right) \mathbf{r}_o - 2 \begin{bmatrix} 0 \\ 0 \\ z_0 \end{bmatrix} \right] \quad (3.13)$$

Higher order orbital oblateness disturbance forces are modeled as described in Appendix D.

Aerodynamic drag on the craft is modeled for each surface of the craft, then summed together to represent the total drag force on the craft. Drag force is quadratically proportional to the ram velocity of the spacecraft, which is calculated assuming an atmosphere with negligible molecular mean thermal motion that is stationary with respect to the rotating surface of the Earth by the below equation:

$$\mathbf{v}_{ram,0} = \mathbf{v}_o - \boldsymbol{\omega}_{Earth} \times \mathbf{r}_o \quad (3.14)$$

where  $\mathbf{v}_{ram,0}$  is calculated in the Earth-centered inertial frame, and

$$\boldsymbol{\omega}_{Earth} = \begin{bmatrix} 0 \\ 0 \\ 7.2921159e-5 \end{bmatrix} rad/s \quad (3.15)$$

For modeling the drag force applied to a single surface, it is also assumed that, per Hughes [44]:

- The linear momentum of impinging gas molecules is completely transferred to the spacecraft surface.

- The linear momentum lost to gas molecules ablated leaving the exposed surface of the spacecraft is negligible.
- The relative velocity associated with spacecraft body rotation for impinging gas molecules is negligible compared to the ram velocity.

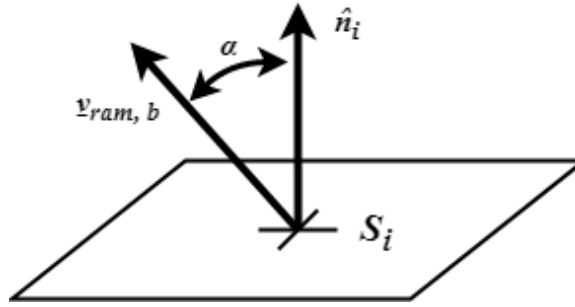
The drag force applied to an individual surface  $S_i$  is calculated in the body frame per the below equation[42]:

$$\mathbf{F}_{drag,b,S_i} = \begin{cases} -\rho \mathbf{v}_{ram,b} \|\mathbf{v}_{ram,b}\| \left( \hat{n}_i \cdot \frac{\mathbf{v}_{ram,b}}{\|\mathbf{v}_{ram,b}\|} \right) A_{s,i}, & \left( \hat{n}_i \cdot \frac{\mathbf{v}_{ram,b}}{\|\mathbf{v}_{ram,b}\|} \right) < 0 \\ 0, & \left( \hat{n}_i \cdot \frac{\mathbf{v}_{ram,b}}{\|\mathbf{v}_{ram,b}\|} \right) \geq 0 \end{cases} \quad (3.16)$$

where

$$\mathbf{v}_{ram,b} = \mathbf{C}_{b,0} \mathbf{v}_{ram,0} \quad (3.17)$$

$\hat{n}_i$  is the normal vector for  $S_i$  that is oriented outwards from the centroid of the solid to which it is attached to the centroid of the surface  $S_i$ , and  $A_{s,i}$  is the cross-sectional area of the surface  $S_i$ . The vectors for a particular surface are illustrated in Figure 3.5. Note that the drag force on a surface element increases proportionally with the cosine of the angle subtended between  $\hat{n}_i$  and  $\mathbf{v}_{ram,b}$ , labelled  $\alpha$  in Figure 3.5.



**Figure 3.5: Surface Element Geometry for Drag Calculation**

The total drag force in the body frame is thus calculated for  $N$  exposed spacecraft surfaces as

$$\mathbf{F}_{drag,b} = \sum_{i=1}^N \mathbf{F}_{drag,b,S_i} \quad (3.18)$$

Acceleration due to drag on the spacecraft is calculated in the inertial frame according to the below equation:

$$\mathbf{a}_{drag} = \frac{\mathbf{C}_{b,0}^{-1} \mathbf{F}_{drag,b}}{m_{s/c}} = \frac{\mathbf{C}_{b,0}^T \mathbf{F}_{drag,b}}{m_{s/c}} = \frac{\mathbf{F}_{drag,0}}{m_{s/c}} \quad (3.19)$$

Atmospheric density is calculated according to the exponential atmosphere model described by Vallado per the below equation [45], where scale height, reference height, and reference density are functions of altitude, defined in Appendix B:

$$\rho(h) = \rho_{ref} e^{-\frac{h-h_{ref}}{H_{scale}}} \quad (3.20)$$

A quaternion describing the orientation of the body frame with respect to the inertial reference frame is defined according to the scalar-first convention:

$$\mathbf{q}_{b0} = \begin{bmatrix} \eta_{b0} \\ \epsilon_{1,b0} \\ \epsilon_{2,b0} \\ \epsilon_{3,b0} \end{bmatrix} = \begin{bmatrix} \eta_{b0} \\ \boldsymbol{\epsilon}_{b0} \end{bmatrix} \quad (3.21)$$

The time rate of change in the quaternion describing spacecraft attitude is calculated per the equation below:

$$\dot{\mathbf{q}}_{b0} = \begin{bmatrix} \dot{\eta}_{b0} \\ \dot{\boldsymbol{\epsilon}}_{b0} \end{bmatrix} = \begin{bmatrix} -\frac{1}{2}(\boldsymbol{\epsilon}_{b0} \cdot \boldsymbol{\omega}_{b0}) \\ \frac{1}{2}((\eta_{b0} \mathbb{I}_{3 \times 3}) \boldsymbol{\omega}_{b0} + \boldsymbol{\epsilon}_{b0} \times \boldsymbol{\omega}_{b0}) \end{bmatrix} \quad (3.22)$$

To describe the change in spacecraft body rates, the below nonlinear dynamics equation is used:

$$\dot{\boldsymbol{\omega}}_{b0} = \mathbf{J}^{-1}(\mathbf{M}_{wheels} + \mathbf{T}_{dist} - \boldsymbol{\omega}_{b0} \times \mathbf{J} \boldsymbol{\omega}_{b0}) \quad (3.23)$$

Disturbance torques accounted for in this model include gravity-gradient disturbances, aerodynamic drag torque, and magnetic dipole effects, per the below equation:

$$\mathbf{T}_{dist} = \mathbf{T}_{drag} + \mathbf{T}_{mag} + \mathbf{T}_{gravity-gradient} + \mathbf{T}_{brake} \quad (3.24)$$

Aerodynamic drag torque was modeled using the results of the above aerodynamic force calculation and the center of pressure to calculate a torque about the center of mass of the spacecraft:

$$\mathbf{T}_{drag} = \mathbf{r}_{com \rightarrow cop} \times \mathbf{F}_{drag,b} \quad (3.25)$$

The aerodynamic center of pressure  $\mathbf{r}_{com \rightarrow cop}$  was calculated according to the below equation:

$$\mathbf{r}_{com \rightarrow cop} = \frac{\sum_{i=1}^N \mathbf{r}_{com \rightarrow centroid,i} \left( \hat{n}_i \cdot \frac{\mathbf{v}_{ram,b}}{\|\mathbf{v}_{ram,b}\|} \right) A_{s,i}}{\sum_{i=1}^N \left( \hat{n}_i \cdot \frac{\mathbf{v}_{ram,b}}{\|\mathbf{v}_{ram,b}\|} \right) A_{s,i}} \quad (3.26)$$

where  $\mathbf{r}_{com \rightarrow cop}$  is the vector from the spacecraft's center of mass to the centroid of surface  $S_i$  with respect to the body frame.

Magnetic dipole-induced torque was modeled using the World Magnetic Model. This model was previously developed in the Horizon Simulation Framework [31] for modeling magnetometer performance. The latest version release of the World Magnetic Model occurred in February 2019, referred to as the WMM2015v2 release. The WMM2015v2 release was used to update the existing model and successfully unit tested for use in the Horizon Simulation Framework. The magnetic field in the ECI frame was calculated for the Julian Date for a given simulation time<sup>1</sup>. Constant values used by the World Magnetic Model are read from an included .COF file by the WMM model in Horizon, then the magnetic field vector in the body frame is calculated based on the asset's Dynamic State as the negative spatial gradient of a scalar potential function of longitude, geocentric latitude, radius, and time, as defined in the equation below [46].

$$\mathbf{b}_{WMM,o}(\lambda, \phi', r, t) = -\nabla V(\lambda, \phi', r, t) \quad (3.27)$$

This potential function is expanded into a series of spherical harmonic terms as

---

<sup>1</sup>A C# DateTime object is used as the time variable for the Horizon WMM model's field calculations, and is calculated based on the Julian Date at a particular simulation time using the method described by Vallado in Algorithm 22 "Inverse Julian Date" [45].



shown in the equation below:

$$V(\lambda, \phi', r, t) = a \sum_{n=1}^N \left(\frac{a}{r}\right)^{n+1} \sum_{m=0}^n (g_n^m(t) \cos(m\lambda) + h_n^m(t) \sin(m\lambda)) \check{P}_n^m(\sin \phi') \quad (3.28)$$

where  $a$  is the reference radius of 6371.2 kilometers,  $g_n^m(t)$  and  $h_n^m(t)$  are time-dependent Gauss coefficients of degree  $n$  and order  $m$  as described in Appendix C, and  $\check{P}_n^m$  are the Schmidt semi-normalized associated Legendre functions defined as

$$\check{P}_n^m(x) = \begin{cases} P_{n,m}(x), & m = 0 \\ \sqrt{2 \frac{(n-m)!}{(n+m)!}} P_{n,m}(x), & m > 0 \end{cases} \quad (3.29)$$

where  $P_{n,m}(x)$  is defined per the below equation [47].

$$P_{n,m}(x) = \frac{1}{2^n n!} (1-x^2)^{m/2} \frac{d^{n+m}}{dx^{n+m}} (x^2-1)^n \quad (3.30)$$

Magnetic torque in the body frame is calculated according to the below equation:

$$\mathbf{T}_{mag} = \mathbf{m}_{dipole} \times \mathbf{b}_{WMM,b} \quad (3.31)$$

where  $\mathbf{b}_{WMM,b}$  is calculated by a frame rotation into the body frame by

$$\mathbf{b}_{WMM,b} = \mathbf{C}_{b,0} \mathbf{b}_{WMM,0} \quad (3.32)$$

Gravity gradient disturbances were modeled using the below formula, where  $\mathbf{r}_b$  is the asset's position in the ECI frame as expressed in body coordinates.

$$\mathbf{T}_{gravity-gradient} = \frac{3\mu}{\|\mathbf{r}_o\|^5} (\mathbf{r}_b \times \mathbf{J} \mathbf{r}_b) \quad (3.33)$$

### 3.4 ADCS Subsystem Model

The ADCS subsystem model is chiefly responsible for implementing the guidance and control to slew the telescope towards a desired attitude, determining the relative attitude error necessary for the system to correct, and the calculating the command torques for reaction wheels to execute pointing maneuvers and control a telescope's

**Table 3.3: Blue Canyon XACT-15 Module Parameters**

Parameter	Value
Pointing Accuracy	+/- 0.003 deg (2 axes) +/- 0.007 deg (3rd axis)
Momentum Capacity	15 mN-m-s
Mass	0.91 kg
Dimensions	10 x 10 x 5 cm
Max Slew Rate	$\geq 10$ deg/s
Lifetime in LEO	5 Years

attitude, as well as any commanded dipole to onboard magnetic torque rods for desaturating reaction wheels using the external torque provided by Earth’s magnetic field.

#### 3.4.1 Linearized PD Attitude Controller

To provide propellant-free coarse pointing control to an individual CubeSat, a proportional-derivative controller architecture was modeled, using an error quaternion and inertial body rates as the states to be controlled. The parameters of the Blue Canyon Technology XACT-15 ADCS module were implemented due to the module’s prior flight history on several prior CubeSat missions, including ASTERIA [3]. A table of parameters for the XACT-15 module can be found in Table 3.3[48].

For providing control torques during nominal mission maneuvers, the Blue Canyon XACT module’s standard 3-axis reaction wheels are utilized. The reaction wheels are Blue Canyon’s RWP-015 series [49], mounted orthogonally within the XACT module enclosure along with a star tracker. Parameters for the RWP-015 are shown in Table 3.4.

To model a linearized proportional-derivative controller for attitude control per De

**Table 3.4: RWP-015 Key Parameters**

Parameter	Value
Momentum Capacity	15 mN-m-s
Max Torque	4 mN-m
Mass	0.130 kg
Dimensions	42 x 42 x 19 mm
Power at Full Momentum Saturation	Less than 1.0 W
Design Life	Greater than 5 years
Moment of Inertia (Spin Axis)	30 kg-mm <sup>2</sup> [50]
Nominal Power	0.5 W[51]
Max Power	5.5 W[51]

Ruiter [42], the process error, in this case the difference between the setpoint attitude and the actual spacecraft attitude modeled as an error quaternion to be minimized can be assumed to be very nearly equal to the desired equilibrium quaternion, namely, the origin quaternion:

$$\mathbf{q}_e \approx 1 + 0\hat{i} + 0\hat{j} + 0\hat{k} \quad (3.34)$$

This quaternion represents perfect alignment between the commanded reference frame and the body reference frame of the spacecraft. The resulting quaternion dynamics can then be modeled as:

$$\dot{\mathbf{q}}_e \approx \begin{bmatrix} \dot{\eta} \\ \dot{\boldsymbol{\epsilon}} \end{bmatrix} = \begin{bmatrix} 0 \\ \pm \frac{1}{2}\boldsymbol{\omega}_e \end{bmatrix} \quad (3.35)$$

With a proportional derivative controller of the form:

$$\mathbf{T}_c = -\mathbf{K}\boldsymbol{\epsilon}_e - \mathbf{K}_d\boldsymbol{\omega}_e \quad (3.36)$$

Substituting the below equation

$$\boldsymbol{\omega}_e = 2\dot{\boldsymbol{\epsilon}}_e \quad (3.37)$$

into Equation 3.23 produces the below non-linear body dynamics equation in the absence of external torques or internal moments:

$$2\ddot{\boldsymbol{\epsilon}}_e = \mathbf{J}^{-1}(\dot{\boldsymbol{\epsilon}}_e \times \mathbf{J}\dot{\boldsymbol{\epsilon}}_e - \mathbf{K}_p\boldsymbol{\epsilon}_e - 2\mathbf{K}_d\dot{\boldsymbol{\epsilon}}_e) \quad (3.38)$$

Linearizing this equation results in the below linear second order differential equation:

$$\ddot{\boldsymbol{\epsilon}}_e = \mathbf{J}^{-1}\left(-\frac{1}{2}\mathbf{K}_p\boldsymbol{\epsilon}_e - \mathbf{K}_d\dot{\boldsymbol{\epsilon}}_e\right) \quad (3.39)$$

Gains can then be selected as for a classical second order linear system controller.

Rewriting the above equation as a second order system of equations

$$\mathbf{J}\ddot{\boldsymbol{\epsilon}}_e + \mathbf{K}_d\dot{\boldsymbol{\epsilon}}_e + \frac{1}{2}\mathbf{K}_p\boldsymbol{\epsilon}_e = \mathbf{0} \quad (3.40)$$

and comparing to the classical example of a homogeneous second-order linear system of ordinary differential equations

$$\ddot{\mathbf{x}} + 2\bar{\zeta}\bar{\omega}\dot{\mathbf{x}} + \bar{\omega}^2\mathbf{x} = \mathbf{0} \quad (3.41)$$

where

$$\bar{\omega} = \begin{bmatrix} \omega_x & 0 & 0 \\ 0 & \omega_y & 0 \\ 0 & 0 & \omega_z \end{bmatrix} \quad (3.42)$$

and

$$\bar{\zeta} = \begin{bmatrix} \zeta_x & 0 & 0 \\ 0 & \zeta_y & 0 \\ 0 & 0 & \zeta_z \end{bmatrix} \quad (3.43)$$

a natural frequency  $\omega$  can be related to  $\mathbf{K}_p$  by the below relation

$$\mathbf{K}_p = 2\mathbf{J}\bar{\omega}\bar{\omega} \quad (3.44)$$

and for  $\mathbf{K}_d$ :

$$\mathbf{K}_d = 2\mathbf{J}\bar{\zeta}\bar{\omega} \quad (3.45)$$

Expressing normalized maximum overshoot as the percentage of the final value by which the controller overshoots the final state for a one-dimensional linearized controller per De Ruiter [42] as

$$M_p = e^{-\pi\zeta\sqrt{1-\zeta^2}} \quad (3.46)$$

and settling time for a convergent system to converge to within 2 percent of its final value as

$$t_s = \frac{\ln 0.02\sqrt{1-\zeta^2}}{-\zeta\omega} \quad (3.47)$$

then controller gains for the linearized system can be calculated for each axis given a settling time specification and a maximum overshoot specification. Assuming that the diagonal elements of  $\bar{\zeta}$  are equivalent for all axes and the same is true for  $\bar{\omega}$  for all body axes, the gains can be calculated as:

$$\mathbf{K}_p = 2\mathbf{J}\mathbb{I}_{3 \times 3}\omega_i^2 \quad (3.48)$$

$$\mathbf{K}_d = 2\mathbf{J}(\mathbb{I}_{3 \times 3}\omega_i)(\mathbb{I}_{3 \times 3}\zeta_i) \quad (3.49)$$

where  $\omega_i$  is equal to the diagonal elements of  $\bar{\omega}$  and  $\zeta_i$  is equal to the diagonal elements of  $\bar{\zeta}$ .

### 3.4.2 LQR Attitude Controller

A linear quadratic regulator controller can also be used in place of a linearized proportional derivative controller to provide full-state feedback attitude control for an asset's attitude and body rates. A linearized state space representation of an asset's attitude dynamics in the absence of disturbance torques is shown in the below continuous state space form:

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u} \quad (3.50)$$

$$\mathbf{y} = \mathbf{C}\mathbf{x} + \mathbf{D}\mathbf{u} \quad (3.51)$$

$$\mathbf{x} = \begin{bmatrix} \epsilon_{1,e} \\ \epsilon_{2,e} \\ \epsilon_{3,e} \\ \omega_x \\ \omega_y \\ \omega_z \end{bmatrix} \quad (3.52)$$

$$\mathbf{u} = \begin{bmatrix} T_{c,x} \\ T_{c,y} \\ T_{c,z} \end{bmatrix} \quad (3.53)$$

$$\mathbf{A} = \begin{bmatrix} \mathbb{O}_{3 \times 3} & \frac{1}{2} \mathbb{I}_{3 \times 3} \\ \mathbb{O}_{3 \times 3} & \mathbb{O}_{3 \times 3} \end{bmatrix} \quad (3.54)$$

$$\mathbf{B} = \begin{bmatrix} \mathbb{O}_{3 \times 3} \\ \mathbf{J}^{-1} \end{bmatrix} \quad (3.55)$$

$$\mathbf{C} = \begin{bmatrix} \mathbb{I}_{3 \times 3} & \mathbb{O}_{3 \times 3} \end{bmatrix} \quad (3.56)$$

$$\mathbf{D} = \begin{bmatrix} \mathbb{O}_{3 \times 3} \end{bmatrix} \quad (3.57)$$

where  $\epsilon_{i,e}$  is the pointing error in the form of the non-real components of an error quaternion about the  $i$ -th axis,  $\omega_i$  is the angular body rate,  $T_{c,i}$  is the control torque about axis  $i$ ,  $\mathbb{O}_{3 \times 3}$  is the three by three matrix of zero elements, and  $\mathbb{I}_{3 \times 3}$  is the three by three identity matrix. Control feedback for providing attitude control is achieved via a full state feedback controller of the form

$$\mathbf{u}_c = -\mathbf{K}\mathbf{x} \quad (3.58)$$

where  $\mathbf{K}$  is the gain matrix to be calculated. The process for developing a linear quadratic regulator and calculating the full state feedback gain matrix entails the development of a control law that minimizes the continuous infinite horizon cost function

below, [52]

$$J_{\infty} = \int_0^{\infty} (\mathbf{z}^T \mathbf{Q} \mathbf{z} + \varrho(\mathbf{u}^T \mathbf{R} \mathbf{u})) dt \quad (3.59)$$

where  $\mathbf{z}^T \mathbf{Q} \mathbf{z}$  and  $\mathbf{u}^T \mathbf{R} \mathbf{u}$  are quadratic time-variant terms proportional to the magnitude of the controlled output and control input, for the linear quadratic regulator, respectively.  $\varrho$  is a constant that proportionally weights the effects of the quadratic terms on the objective function, allowing tradeoffs between minimizing controlled output or control input. By default, a  $\varrho$  value equal to 1 results in a balanced weighting between minimizing control input and controlled output.  $\mathbf{z}(t)$  is defined per the below equation:

$$\mathbf{z}(t) = \mathbf{G}\mathbf{x}(t) + \mathbf{H}\mathbf{u}(t) \quad (3.60)$$

where  $\mathbf{G}$  relates the full state to the controlled output  $\mathbf{z}$  and  $\mathbf{H}$  relates the control input  $\mathbf{u}$  to the controlled output.  $\mathbf{Q}$  and  $\mathbf{R}$  can be initially selected using Bryson's Rule, such that

$$\mathbf{Q}_{ii} = \frac{1}{(z_i^*)^2} \quad (3.61)$$

and

$$\mathbf{R}_{ii} = \frac{1}{(u_i^*)^2} \quad (3.62)$$

where  $z_i^*$  is the maximum allowable absolute value of the  $i$ -th component of  $\mathbf{z}$  and  $u_i^*$  is the maximum allowable absolute value of the  $i$ -th component of  $\mathbf{u}$ . Subsequently, the gain matrix  $\mathbf{K}$  can be calculated using the solution to the algebraic Riccati equation for the above linear time invariant system as

$$\mathbf{K} = (\mathbf{H}^T \mathbf{Q} \mathbf{H} + \varrho \mathbf{R})^{-1} (\mathbf{B}^T \mathbf{P} + \mathbf{H}^T \mathbf{Q} \mathbf{G}) \quad (3.63)$$

where  $\mathbf{P}$  is the positive-definite solution to the Algebraic Ricatti Equation of the form

$$\mathbf{A}^T \mathbf{P} + \mathbf{P} \mathbf{A} + \mathbf{G}^T \mathbf{Q} \mathbf{G} - (\mathbf{P} \mathbf{B} + \mathbf{G}^T \mathbf{Q} \mathbf{H}) (\mathbf{H}^T \mathbf{Q} \mathbf{H} + \varrho \mathbf{R})^{-1} (\mathbf{B}^T \mathbf{P} + \mathbf{H}^T \mathbf{Q} \mathbf{G}) = 0 \quad (3.64)$$

For developing an LQR-designed PD attitude controller for the CubeSat Astronomy Network spacecraft, the following parameters were used:

$$\mathbf{Q} = \begin{bmatrix} (2.613)\mathbb{I}_{3 \times 3} & \mathbb{O}_{3 \times 3} \\ \mathbb{O}_{3 \times 3} & \mathbb{O}_{3 \times 3} \end{bmatrix} \quad (3.65)$$

$$\mathbf{Q} = \begin{bmatrix} (\frac{1}{.004})\mathbb{I}_{3 \times 3} \end{bmatrix} \quad (3.66)$$

Using MATLAB's `lqr()` command with the above parameters, including those of Equation 3.54 and Equation 3.55 resulted in the following calculated gains:

$$\mathbf{K}_p = \begin{bmatrix} (0.0065)\mathbb{I}_{3 \times 3} \end{bmatrix} \quad (3.67)$$

$$\mathbf{K}_d = \begin{bmatrix} (0.0311)\mathbb{I}_{3 \times 3} \end{bmatrix} \quad (3.68)$$

Linear step response parameters were calculated and are shown in Table 3.5.

**Table 3.5: LQR Controller Step Response Parameters**

Parameter	Value
Rise Time [s]	14.6
Settling Time [s]	40.6
Overshoot	4.32%

### 3.4.3 Reaction Wheel Momentum Desaturation Controller

To desaturate onboard reaction without requiring the use of expendable onboard propellant, three orthogonally-mounted magnetic torque rods, which are also commonly referred to as magnetorquers, are used to induce oppositenal moment to the braking torque applied to the reaction wheels during a desaturation maneuver. Desaturation maneuvers are used to reduce the total accumulated momentum in the wheels after a series of slew maneuvers, as extended operation at high wheel speeds



near their rated momentum capacity can induce undesirable jitter during imaging operations and potentially reduce their operational lifetimes.

To model the performance of desaturation activities during mission operations for the CubeSat Astronomy Network, the onboard magnetic torque rods were used as the primary actuators in the linearized PD control loop, calculating the quaternion error with respect to the commanded attitude to determine a control torque. This control torque was then used to calculate a dipole command to be executed by the magnetic torque rods, per the cross product control law below per Tregouet et al. [53]:

$$\mathbf{m}_{dipole,comm} = -\mathbf{b}_{wmm,b} \times \mathbf{T}_{control} \quad (3.69)$$

This commanded dipole then produced a magnetic torque on the spacecraft, enabling the craft to maintain a degree of attitude stability while applying a braking torque to the reaction wheels. A braking torque on the reaction wheels was then generated according to the below equation, proportional to the dipole torque produced by the magnetic torque rods.

$$\mathbf{T}_{brake} = \mathbf{K}_{brake} sgn(\omega_{wb}) abs(\mathbf{T}_{control}) \quad (3.70)$$

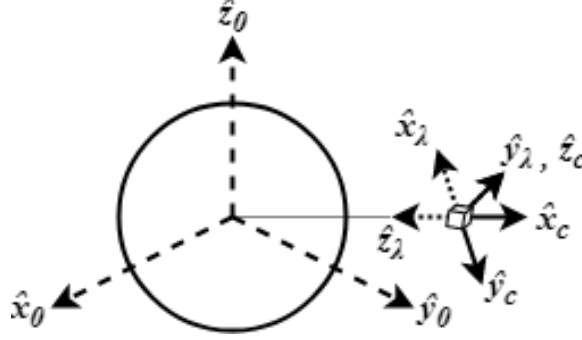
During this maneuver, the commanded attitude aligned the spacecraft axes with the local horizontal reference frame per the relation described in subsection 3.4.4.

#### 3.4.4 Commanded Attitude Pointing Modes

There are three principal attitude command frames specified for the CubeSat Astronomy Network during the execution of system tasks. The command frame is a reference frame with which the body axes are commanded to align, analogous to a setpoint value in classical control systems. A command frame can be defined per the vectrix below:

$$\vec{\mathcal{F}}_c = \begin{bmatrix} \hat{\mathbf{x}}_c & \hat{\mathbf{y}}_c & \hat{\mathbf{z}}_c \end{bmatrix}^T \quad (3.71)$$

These command frames correspond to three pointing operational modes: idle or desaturation pointing, imaging pointing, and downlink pointing. For idle or desaturation pointing, the spacecraft body axes are commanded to align as follows:  $\hat{\mathbf{x}}_c$  is aligned with  $-\hat{\mathbf{z}}_\lambda$ ,  $\hat{\mathbf{y}}_c$  is aligned with  $-\hat{\mathbf{x}}_\lambda$ , and the commanded boresight axis  $\hat{\mathbf{z}}_c$  is aligned with  $\hat{\mathbf{y}}_\lambda$ . This is illustrated in



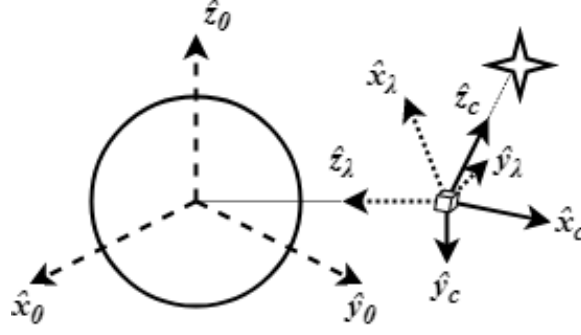
**Figure 3.6: Idle and Desaturation Pointing Mode**

For imaging targets, the boresight axis  $\hat{\mathbf{z}}_c$  is commanded to align in the direction of the target in the ECI frame, and the  $\hat{\mathbf{x}}_c$  axis is aligned with the projection of the zenith vector in the local horizontal frame onto the normal plane of the boresight axis, according to the formula:

$$\hat{\mathbf{x}}_c = -\hat{\mathbf{z}}_\lambda + \hat{\mathbf{z}}_c(\hat{\mathbf{z}}_c \cdot \hat{\mathbf{z}}_\lambda) \quad (3.72)$$

$\hat{\mathbf{y}}_c$  completes the frame by the right-hand rule. Once this attitude has been achieved by the body frame, the spacecraft then commands it to hold inertial pointing until the commanded dwell time duration has passed: then the spacecraft can slew to a new attitude, or hold the current one as long as pointing constraints are satisfied. This pointing mode is illustrated in Figure 3.7.

For downlinking science data to a Ground Segment facility, the antenna-pointing axis  $-\hat{\mathbf{x}}_b$  is commanded to align in the direction of the facility  $-\hat{\mathbf{x}}_c$ , and the commanded boresight axis  $\hat{\mathbf{z}}_c$  is aligned with the projection of  $\hat{\mathbf{y}}_\lambda$  onto the normal plane

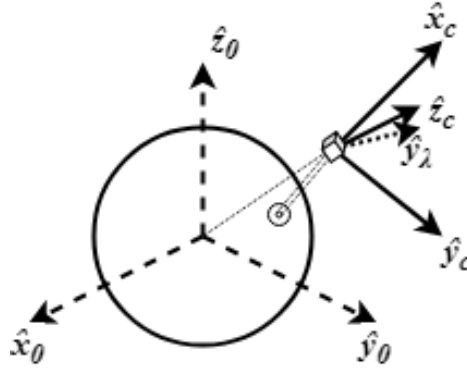


**Figure 3.7: Imaging Pointing Mode**

of  $-\hat{\mathbf{x}}_c$  according to the formula:

$$\hat{\mathbf{z}}_c = \hat{\mathbf{y}}_\lambda - \hat{\mathbf{x}}_c(\hat{\mathbf{x}}_c \cdot \hat{\mathbf{y}}_\lambda) \quad (3.73)$$

$\mathbf{y}_c$  then completes the command frame by the right hand rule. This pointing mode is illustrated in Figure 3.8. Tracking the ground target during the pass is necessary to minimize pointing losses during a communication pass over a ground station.



**Figure 3.8: Ground Station Communication Pointing Mode**

### 3.5 Payload Subsystem Model

For modeling the performance of the imaging payload, a model template made for testing the Aeolus test case's scripting functionality was adapted for modeling the mission data throughput and operating conditions for each telescope. Analysis of off-the-shelf sensors suitable for these applications led to further investigation of Teledyne

e2v’s CCD47-10 NIMO sensor and the 3DCM734-1 CMOS camera offered by 3D Plus [4]. For the purposes of modeling science data throughput, an assumed sensor format of 1024 by 1024 bixels, with a pixel depth of 16 bits per pixel, was used for estimating the size of full-frame science images. Larger format, lower pixel bit resolution sensors could utilize pixel binning or frame stacking to achieve equivalent data throughput, though this data volume can be specified by a user in the XML model file. This sensor configuration was assumed due to early-identified system sensitivities to onboard science data generation, and the relatively limited throughput of traditional CubeSat RF communications. Sub-frame science image volume is modeled as a 128 by 128 pixel window containing the observation target, and the gathering of full-frame or sub-frame image data can be specified within the XML target file by ascribing a different target value to an imaging target.

### 3.6 Mission Data Handling Subsystem Model

The mission data handling subsystem models the generation, compression, storage, and transmission of mission data products for an individual asset. During an imaging task, the subsystem models the generation and compression of image data recorded during payload operation, and calculates the resulting data buffer fill ratio for the onboard storage capacity after the imaging task. If the data buffer capacity is exceeded, then the subsystem’s `canPerform()` method returns `False` due to the inability of the system to record data for the entire duration of the imaging task. The image data for an observation is modeled after compression by applying a proportional compression ratio to the image data volume. Lossless image compression algorithms for 16 bit imagers, as tested by Pence et al. [54] based on FITS tiled-image compression using the Rice algorithm, are capable of achieving compression ratios of 1.7 or greater for deep exposures with times on the order of 500 to 600 seconds. Conservatively, a worst-case scenario compression ratio of 1.5 is assumed to be an average image com-

pression ratio for the CubeSat Astronomy Network. Key parameters for the Mission Data Handling subsystem are shown in Table 3.6. For modeling the transmission of

**Table 3.6: Mission Data Handling Subsystem Parameters**

Parameter	Value
Science Data Compression Ratio	1.5
Science Data Storage	8 GB

science data during a pass over a ground station, the subsystem models the removal of data from the storage buffer during the maneuver, and calculates the required data rate profile during the downlink pass to transmit continuously during the duration of the pass.

### 3.7 Communications Subsystem Model

For modeling the performance of the Communications subsystem for the CubeSat Astronomy Network space assets, the performance of a downlink transmission is modeled by calculating the relative elevation angle of the space asset in the ground station’s topocentric reference frame: if the elevation angle is calculated to be less than the minimum acceptable elevation angle to successfully transmit the data with sufficient link margin, as calculated by the user, then the communication task is not successfully executed by the system. Key parameters for the communications subsystem are shown in Table 3.7. A design tool commonly leveraged for non-commercial

**Table 3.7: Communication Subsystem Key Parameters**

Parameter	Value
Minimum Ground Station Elevation Angle during Downlink	10 degrees
Peak Data Rate	1.67 Mbit/s

CubeSat mission design for link budget analysis is the AMSAT IARU Link Model [55]. The Communications subsystem model includes interoperability with Microsoft Excel to allow the communications bit rate value calculated in this AMSAT spreadsheet to be used as the peak data rate for the subsystem, demonstrating a potential interface for future exploration in subsequent versions of the Horizon Simulation Framework. Existing spreadsheet-based toolsets for system design and analysis could be directly referenced by Horizon to either generate input files or subsystem attributes in the framework, improving the user operability for utilizing the framework for model-based systems engineering applications. This AMSAT link model was used to determine a satisfactory data rate for modeling downlink operations, and is shown in Appendix E. Most values in this model spreadsheet were derived from prior design work in assessing the feasibility of the CubeSat Astronomy Network [4], or assumed based on prior CubeSat mission link budgets utilizing the Cal Poly ground station facility.

### 3.8 Power Subsystem Model

For modeling the charging and discharging of the electrical power system via solar panel and power dissipation of onboard components, the existing power subsystem model from the Aeolus test case code was modified to include additional design parameters for modeling the change in system state, specifically incoming solar power and depth of discharge, for the CubeSat Astronomy Network spacecraft. Rate of change in depth of discharge is modeled as:

$$D\dot{O}D = \frac{P_{out} - P_{in}}{E_{battery}} \quad (3.74)$$

where  $P_{out}$  is the power consumed by all subsystems in Watts,  $P_{in}$  is the input solar power in Watts, and  $E_{battery}$  is the total power capacity of the onboard batteries.

Input solar power is modeled using the below equation:

$$P_{in} = 1367 A_{panel} \eta_{panel} \rho_{cell} (\hat{n}_{panel} \cdot \hat{r}_{s/c \rightarrow sun}) \quad (3.75)$$

where  $A_{panel}$  is the total solar panel area on the spacecraft's -Z face,  $\eta_{panel}$  is the equivalent solar panel efficiency,  $\rho_{cell}$  is the solar cell collecting area density on the solar panel such that

$$\rho_{cell} = \frac{\sum_{i=1}^N A_{cell,i}}{A_{panel}} \quad (3.76)$$

and  $\hat{n}_{panel} \cdot \hat{r}_{s/c \rightarrow sun}$  is the dot product between the panel's sun-facing normal vector and the normal vector from the spacecraft center of mass to the Sun. Key parameters for the power subsystem are shown in Table 3.8, and were validated as part of the demonstrated results of simulation work. A more detailed conceptual power budget is shown in Appendix F.

**Table 3.8: Power Subsystem Key Parameters**

Parameter	Value
Panel Area	0.18 $m^2$
Panel Efficiency	25%
Cell Density	75%
Thermal Control Power Req.	12 W

### 3.9 Design Reference Mission

To characterize the performance of the CubeSat Astronomy Network, a baseline mission representing a typical use case for the Network's day-in-the-life operations was developed. Two variants of this mission were designed to be compared: a single-asset mission composed of a single space telescope, a ground station, and an observation target; and a multiple-asset mission, consisting of a constellation of three telescopes,

**Table 3.9: CubeSat Astronomy Network Reference Mission Targets**

Target	Coordinates Type	Units	Coordinates
Ground Station	Latitude	Degrees	35.30
	Longitude	Degrees	-120.67
	Altitude	Meters	75
Imaging Target	Right Asc.	Degrees	83.3449
	Declination	Degrees	1.764
	Distance	Light-years	35.65

a ground station, and an observation target. For modeling target observation performance, the inertial coordinates of Beta Virginis were used. Beta Virginis was selected as a reference mission target due to its well-characterized photometric attributes, making it valuable for instrument calibration during mission operations in the month of March. A desaturation target location is also included in the anti-solar direction, but the commanded attitude during a desaturation maneuver is simply the idle local-nadir aligned frame with the boresight axis pointed in the nearly anti-solar direction. Parameters for the ground station and observation target are shown in Table 3.9:

The simulated time domain for the reference mission begins at 00:00:00.0 UTC on March 12, 2021. A date near the time of the vernal equinox that also coincided with minimal chance of target exclusion was chosen to aid in debugging the model. Future analysis over longer simulated time domains will need to account for the effects of lunar interference on scheduling observations.

Per the requirement to perform time series photometric requirements of science targets, and given the prior mission heritage of similar missions such as MOST [56] and BRITe [28], a sun-synchronous orbit at no less than 500 kilometers altitude with



local crossing times of 6 AM and 6 PM was selected for satisfying mission requirements for the CubeSat Astronomy Network.



### 3.9.1 Single Satellite Reference Mission

For modeling and simulating the performance of CubeSat Astronomy Network mission activities by a single spacecraft, the initial orbital parameters described in Table 3.10 were used to model and simulate the spacecraft’s dynamic state during execution of mission tasks. The asset begins with zero angular rates and zero wheel speeds, and is initially aligned with the local horizontal frame in the commanded-nadir pointing mode described in Figure 3.6.

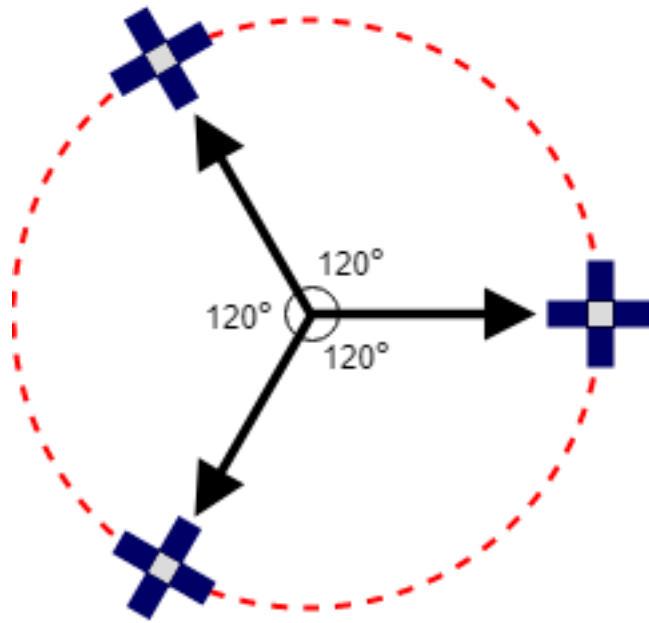
**Table 3.10: Single Satellite Orbit Initial Conditions**

Parameter	Value
Semi-Major Axis [km]	6878.0
Eccentricity	0.005
Inclination [degrees]	97.4023
Right Ascension of Asc. Node [degrees]	82.05
Argument of Perigee [degrees]	0.0
True Anomaly [degrees]	0.0

### 3.9.2 Constellation Reference Mission

For modeling the performance of the CubeSat Astronomy Network in a constellation configuration, three equally-spaced space assets in a coplanar orbit identical to the above orbital plane for the single-asset use case were modeled and simulated in a constellation reference mission. More space assets could be theoretically modeled, but the scheduling algorithm’s time complexity varies exponentially with the number of simulated assets: thus, exponentially greater amounts of run time would be required

to simulate the performance of constellations with greater numbers of satellites. For demonstrating the feasibility of using the Horizon Simulation Framework to assess the feasibility of using a small number of satellites to achieve near-constant imaging coverage of an observation target, a constellation of three satellites at the most was deemed satisfactory for this purpose. The initial orbital conditions of the constellation assets are shown in Table 3.11. The diagram illustration in Figure 3.9 shows the relative spacing between each satellite in the orbital plane. The initial orbital



**Figure 3.9: CubeSat Astronomy Network Constellation Configuration**

conditions for the orbital plane are identical to those for the single satellite mission. Each satellite begins with zero angular rates, zero reaction wheel speeds, and aligned per the commanded nadir-pointing mode illustrated in Figure 3.6.

**Table 3.11: Constellation Reference Orbit Initial Conditions**

Parameter	Value
Semi-Major Axis [km]	6878.0
Eccentricity	0.005
Inclination [degrees]	97.4023
Right Ascension of Asc. Node [degrees]	82.05
Argument of Perigee [degrees]	0.0
True Anomaly [degrees]	0.0 [1], 120.0 [2], 240.0 [3]

## Chapter 4

### ANALYSIS

Per the analytical relations described in chapter 3, the simulated results of operation after 6000 seconds of simulated time are described as follows. The single satellite mission results are presented and briefly described. Similarly, the constellation mission results are presented and described.

#### 4.1 Summary of Simulation Results

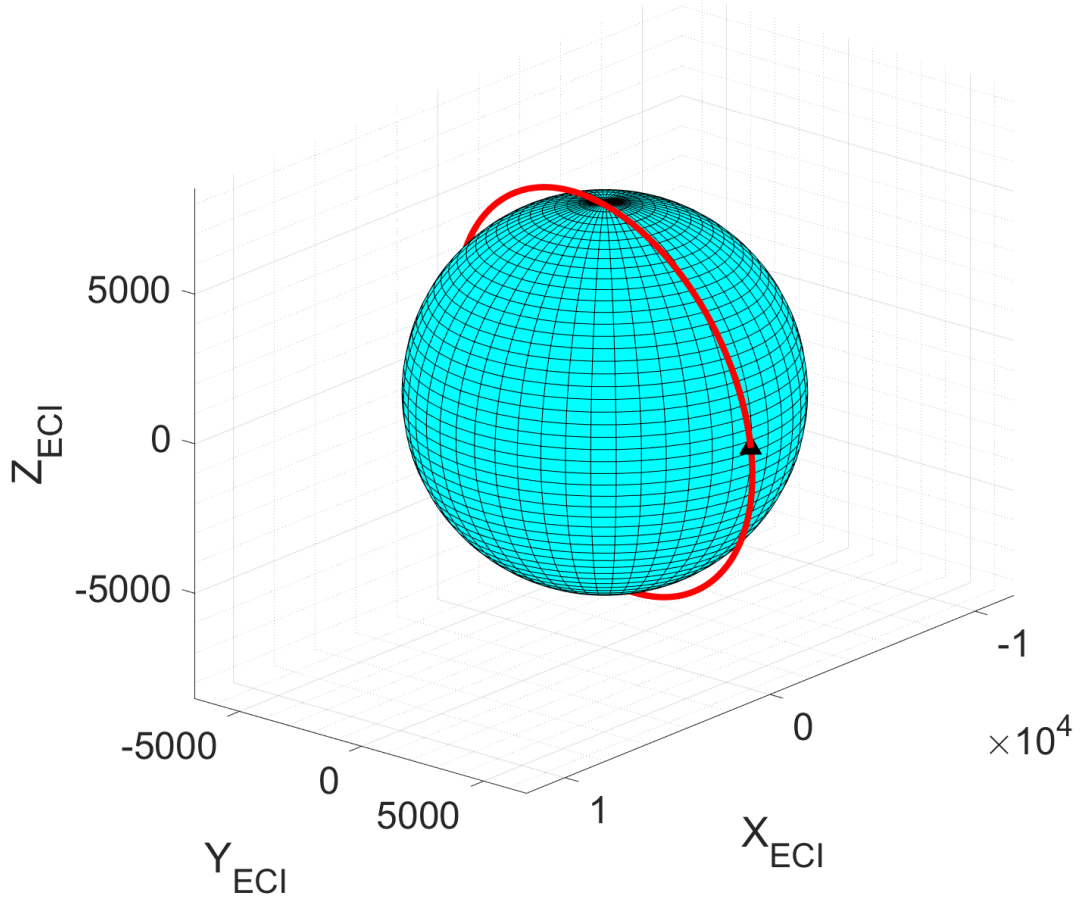
The simulations performed of the single satellite and constellation reference missions were able to successfully generate schedules that satisfied the system constraints derived from system-level requirements. Continuous observation capabilities were demonstrated by both simulations, with interruptions between observations in the case of the single satellite mission and uninterrupted target coverage capabilities in the case of the constellation mission. Further design work using higher fidelity models will be necessary to better assess the feasibility of the system concept, but the results from using the Horizon Simulation Framework can inform that development as well as the necessary design considerations. The output state data information of each mission case is presented in the sections below.

#### 4.2 Single Satellite Case Results

The simulated results for the single satellite reference mission using the Horizon Simulation Framework are shown in the following figures. The orbital trajectory and starting position of the satellite in the Earth-centered inertial reference frame are shown in Figure 4.1.

The spacecraft's attitude with respect to the inertial frame over time, represented as a quaternion time series, is shown in Figure 4.2. Note the segments of ramping

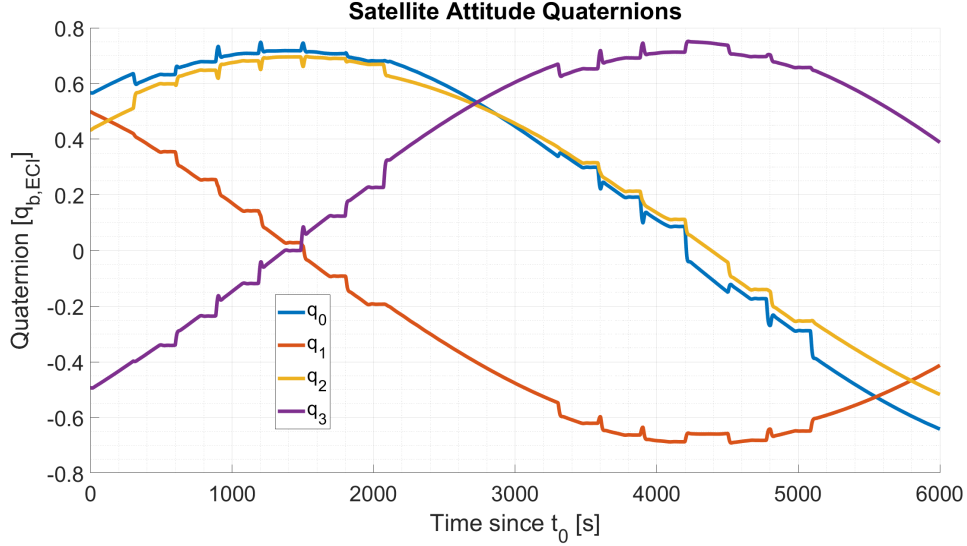
## CAN Single Satellite Orbit



**Figure 4.1: Single Satellite Mission Orbit Geometry**

and steady-state quaternion values during imaging operations, as the spacecraft slews to a new orientation with each imaging operation to align itself with the horizon to provide stray light protection from potential stray light from the Earth's illuminated limb.

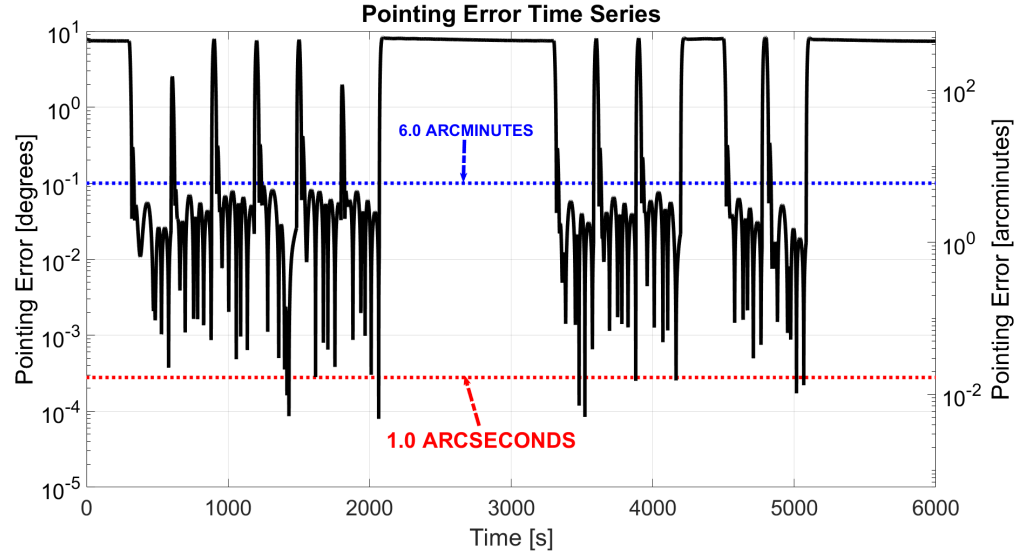
The pointing error between the spacecraft boresight axis and the imaging target is plotted over the duration of the simulation in Figure 4.3. Note that during imaging maneuvers, the attitude pointing error rarely falls below 1 arcsecond of pointing error, but lies within 6 arcminutes, or 0.1 degrees during the duration of an individual imaging task. While this pointing capability is not representative of the documented



**Figure 4.2: Single Satellite Mission Attitude Time Series**

capabilities of the Blue Canyon XACT module, other missions that have leveraged the XACT, such as ASTERIA and the proposed mDOT mission [57], leverage a two-stage fine pointing method for their imaging payloads. In ASTERIA's case, the XACT module is used to provide coarse pointing capability, and active piezoelectric actuators at the imaging plane provide image stabilization, enabling the equivalent of sub-arcsecond pointing capability for the payload. The mDOT mission proposes the use of an actively-actuated tip-tilt mirror with a mechanical range of  $\pm 1$  degree to provide fine image stabilization for the imaging payload. A similar two-stage approach to providing fine pointing capability for the imaging payload for the CubeSat Astronomy Network spacecraft ought to be considered during future imaging payload design to ensure data fidelity is not compromised during imaging operations due to the limited pointing capabilities of the spacecraft bus. An active tip-tilt mirror with the Mirrorcle Technologies S6180 TT mirror as described by Beierle et al. [57] for the mDOT payload may have sufficient stroke range to correct the attitude pointing error of the CubeSat Astronomy Network spacecraft.

The spacecraft body angular rates are shown in Figure 4.4. Note that the rates

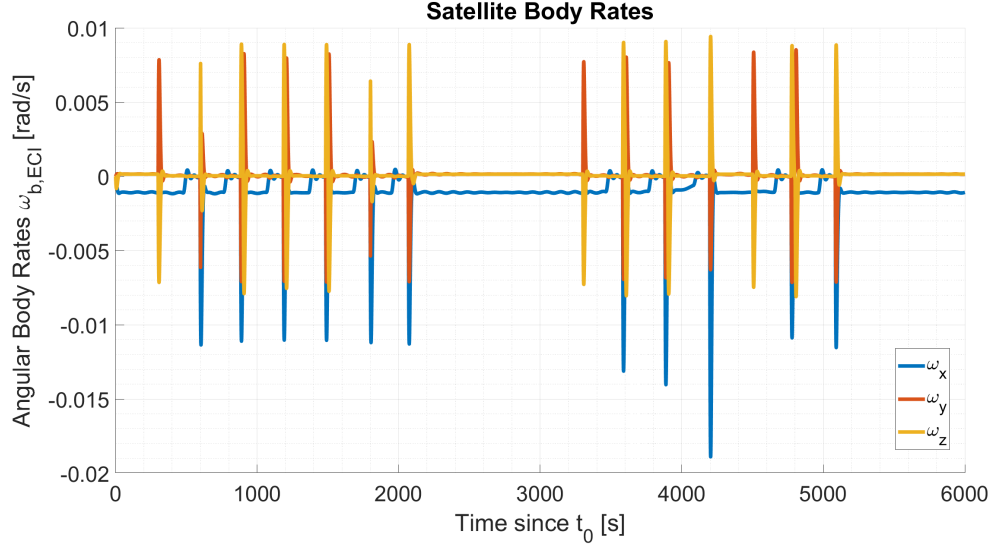


**Figure 4.3: Single Satellite Mission Pointing Error Time Series**

about all axes do not exceed a magnitude of 0.02 radians per second, or a little greater than 1 degree per second.

The spacecraft's RWP-015 reaction wheel speeds over time are shown in Figure 4.5. Note the secular trends in the reaction wheel speeds towards their maximum values. It is this secular trend in reaction wheel speeds towards their maximum rated speeds that necessitates the desaturation of reaction wheels during the course of nominal mission operations. This can prevent or minimize undesired wear or damage due to excessive wheel vibration at high operating speeds, thus improving the components' operational lifetime and minimizing the risk of mission-critical system failure early in the mission.

To estimate the frequency of reaction wheel desaturation during nominal mission operations, linear least squares regression was used to estimate the secular trend of each reaction wheel's speed during the course of one orbit using the wheel speed data. A 95% prediction interval of each linear trend was also calculated using MATLAB's `polyfit()` and `polyval()` functions. The linear trends were then used to extrapolate estimated time required for each wheel's 95% predictive interval to reach 80% of its



**Figure 4.4: Single Satellite Mission Body Rates Time Series**

maximum rated speed of 500 radians per second. The extrapolated times were then used to calculate the 95% prediction interval at the extrapolated saturation time to assess the validity of the extrapolated values. The data of the linear trends is plotted in Figure 4.6. Per the results of the linear trend analysis, it is estimated that the reaction wheels may require desaturation after no less than 9.5 hours during an imaging mission, assuming the wheels start with nearly-zero speeds at the end of desaturation. However, beginning imaging operations with negligible initial wheel speeds may pose the risk of affecting pointing precision, as the ASTERIA mission found that when each axis' reaction wheel speed is reversed, per Pong, "the resulting error transient is observed on the expected axis [3]". These transients are possibly due to static friction or other mechanical disturbances characteristic of the reaction wheel bearings, which may be especially detrimental to CubeSat sized spacecraft that need to maintain precise pointing over extended periods of time. To overcome the problems of transient pointing error effects during reaction wheel zero-speed crossings, Pong describes how "reaction wheel speed reversals can be avoided during observations by commanding the XACT to have a momentum bias in the inertial frame such that the reaction wheel





**Figure 4.5: Single Satellite Mission Wheel Speeds Time Series**

speeds are all, for example, 1000 rpm while in a particular inertially-fixed frame”. A similar approach may be possible for the CubeSat Astronomy Network spacecraft to supply fine pointing capability as part of the scheduling approach: simulations performed as part of Ground Segment mission operations could assess the required bias speeds to maintain steady pointing during an observation. If a successful schedule can be generated, the observation and mission command schedule can be uploaded to the spacecraft to execute the required maneuvers and system tasks. However, as Pong elaborates, the exact command biases must be carefully chosen as commanding larger momentum biases ”increases the chance [of] saturating the reaction wheels, which would also cause significant pointing errors.” Thus, future development of a reaction wheel momentum biasing methodology may be required in future design work for the CubeSat Astronomy Network.

To assess the feasibility of the reaction wheel desaturation method described in chapter 3, a shorter satellite simulation was developed with initial conditions identical to the single satellite mission simulation conditions, with the exception of the initial y-axis wheel speed being equal to 80% of the maximum rated wheel speed. The results

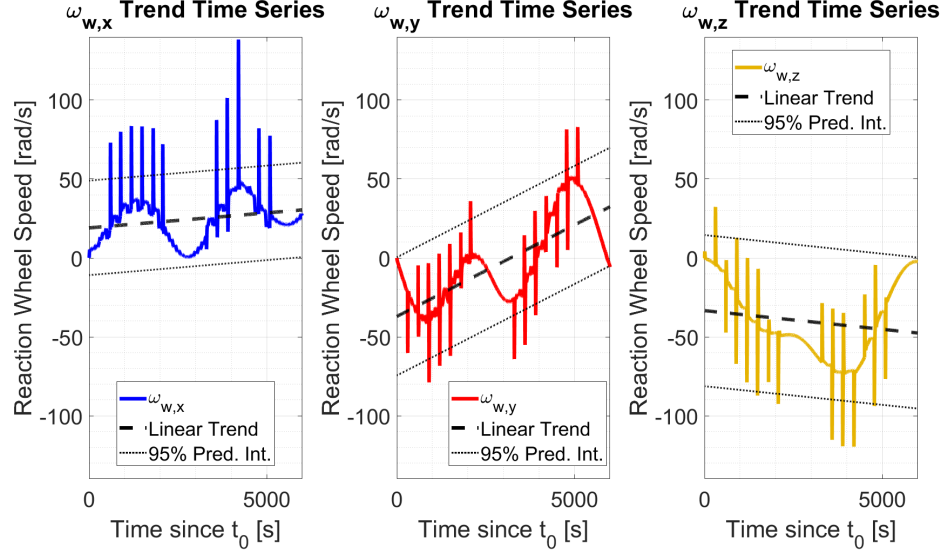


Figure 4.6: Single Satellite Wheel Trends Time Series

Table 4.1: Reaction Wheel Saturation Trend Analysis

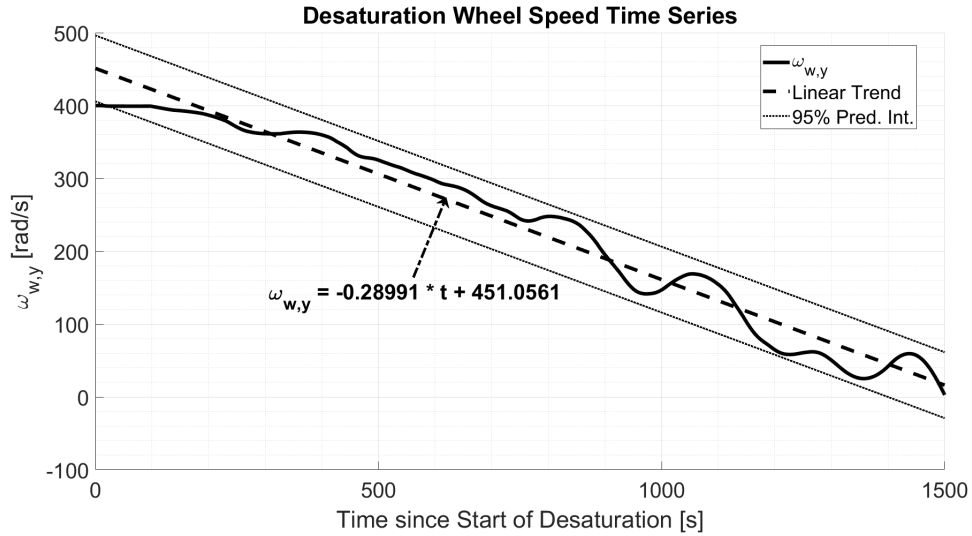
Wheel	Linear Least Squares Best Fit [rad/s]	Mean Std. Error [rad/s]	Extrapolated Sat. Time [s]
X	$0.00190t + 19.038$	14.962	184821.888
Y	$0.0116t + -37.068$	18.708	34475.949
Z	$-0.00236t + -33.366$	23.931	175407.710

of the simulation are shown in Figure 4.7, showing the capability to desaturate a single wheel from the maximum allowable speed in no less than 1500 seconds, or 25 minutes. However, a more robust control design may be needed to ensure compliance with spacecraft pointing constraints, and the limited acutation capabilities of the iMTQ board magnetorquers, in addition to the limitations of magnetoquener control due to the cross-product dependence on the magnetic field, which removes a degree of controllability from the attitude control system without relying on another momentum exchange device, propulsive attitude actuator, or some other means of generating a control torque. However, operation in a nearly-polar orbit can mitigate this problem

**Table 4.2: Reaction Wheel Saturation Trend Analysis, Continued**

Wheel	Extrapolated Wheel Speed [rad/s]	Extrapolated Std. Error [rad/s]	Extrapolated 95% Wheel Speed Pred. Interval
X	370.075	25.193	$370.075 \pm 50.385$
Y	362.583	19.214	$362.583 \pm 38.429$
Z	-447.861	38.955	$-447.862 \pm 77.910$

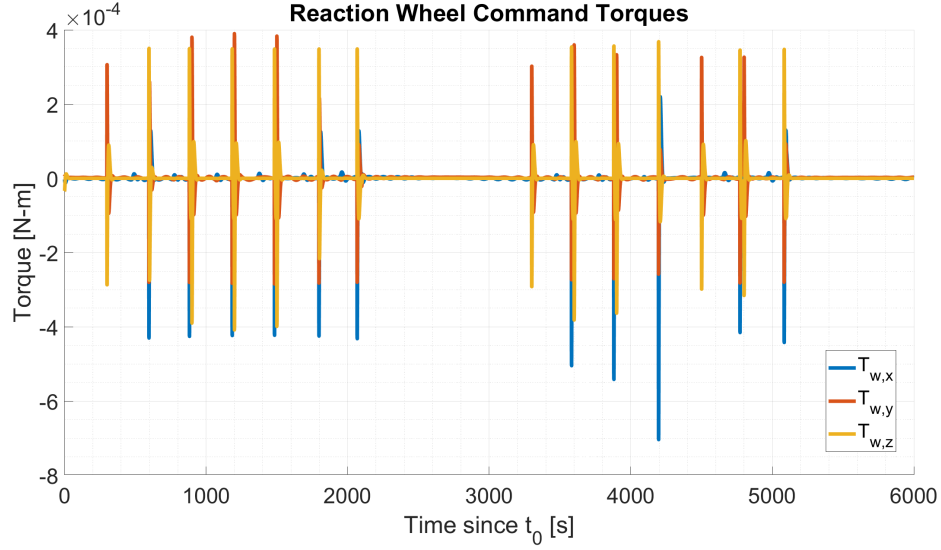
slightly, as the magnetic field lines in the inertial reference frame will change in magnitude and direction at different points in the spacecraft's orbit, especially near the polar regions. Further dedicated attitude control system design will need to consider these factors in future design work.



**Figure 4.7: Single Satellite Wheel Desaturation Time Series and Trend**

The commanded torques required to provide spacecraft attitude control are shown in Figure 4.8. Note that the magnitudes of these torques do not exceed the maximum rated torque of the RWP-015, which is 0.004 N-m.

The relative angle subtended between the boresight axis of the payload and the



**Figure 4.8: Single Satellite Mission Command Torque Time Series**

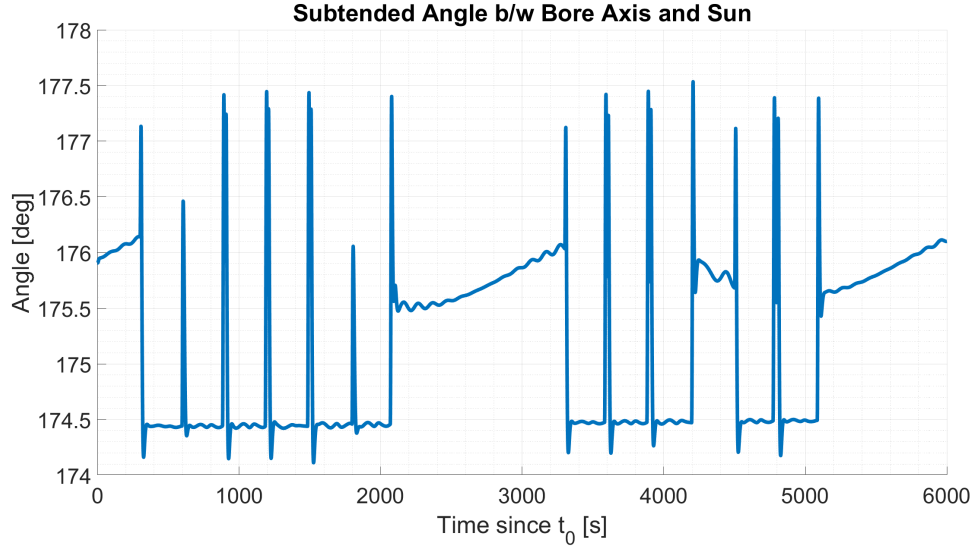
Sun is plotted in Figure 4.9. Due to the relative orientation of the boresight axis to the solar panels, an angle greater than 135 degrees ensures that the solar panels are never pointed more than 45 degrees away from the Sun.

The relative angle subtended between the boresight axis of the payload and the velocity vector of the spacecraft is plotted in Figure 4.10. Note that the angle is always greater than 75 degrees.

The relative angle subtended between the boresight axis of the payload and the local nadir vector of the spacecraft is plotted in Figure 4.11. Note that the angle is always greater than 75 degrees.

The times and number of bytes of raw image data recorded by the payload and data handling system are shown in Figure 4.12. The individual values correlate to 16-bit, 1024 by 1024 pixel, monochromatic image data per imaging task.

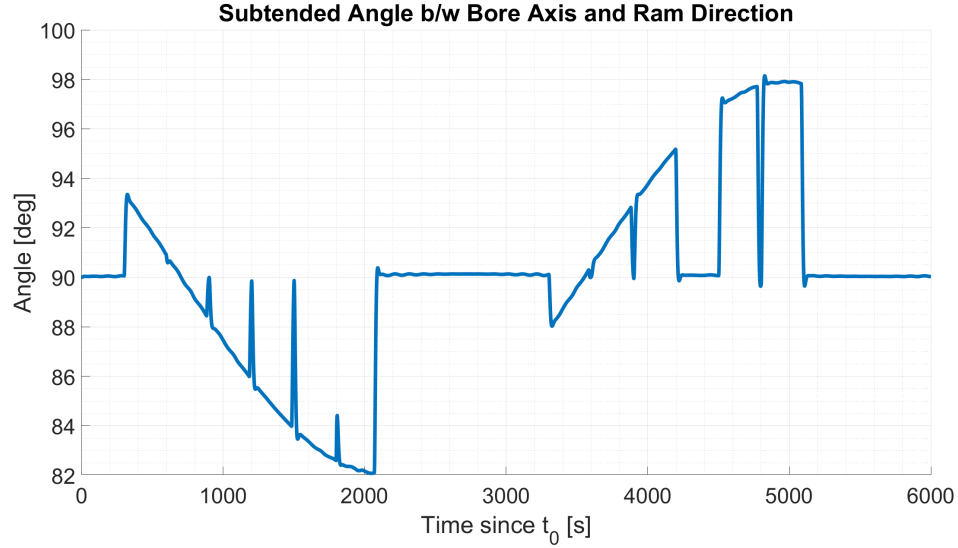
The amount of onboard storage utilized by the image data after compression (with a modeled Rice compression ratio of 1.5) is plotted in Figure 4.13. Note that with storage capacity on the order of gigabytes, greater data storage density can prove riskier due to the more detrimental effects of radiation damage on smaller electronic



**Figure 4.9: Single Satellite Mission Sun Angle Time Series**

elements such as transistors and memory registers. Therefore, the benefits of greater storage density against robustness against radiation damage must be carefully evaluated during preliminary and detailed subsystem design.

To estimate the amount of operations time required before the onboard memory completely filled, the secular change in onboard science data memory plotted in Figure 4.13 was characterized using linear least squares regression. This linear fit, plotted in Figure 4.14, was then used to extrapolate the estimated time when the maximum onboard data constraint would be violated by the trend's 95% predictive interval. The extrapolated data volume and standard error were then calculated according to the linear model at the extrapolated time to assess the validity of the extrapolated value. As shown in the linear trend, it is estimated that the entire data volume of 8GB of science data onboard memory could be utilized to 85% capacity in no greater than roughly 12 days. Thus, a decreased amount of onboard data storage may be sufficient for successful execution of mission activities. However, it is still critical to downlink the science data quickly enough to avoid exceeding onboard memory capacity. From an operations standpoint, this could possibly be addressed by prioritizing



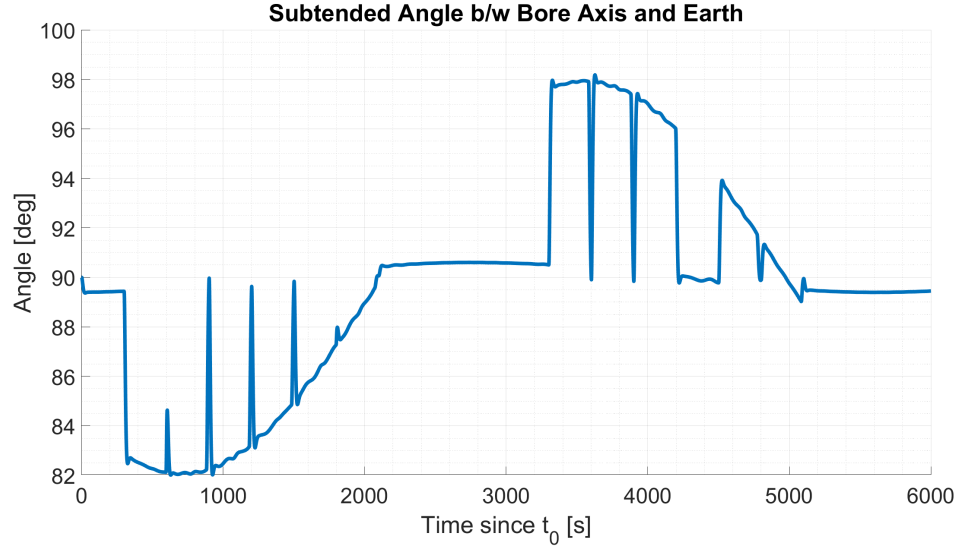
**Figure 4.10: Single Satellite Mission Ram Angle Time Series**

mission data downlink only after a desired observation window has passed, at which point downlinking and spacecraft health checks could be purely scheduled. Further analysis may be required to consider the tradeoffs of a "store and forward" downlink schedule based purely on ground station availability, or intentionally delayed until after a scheduled observation series has concluded.

**Table 4.3: Data Storage Utilization Trend Analysis**

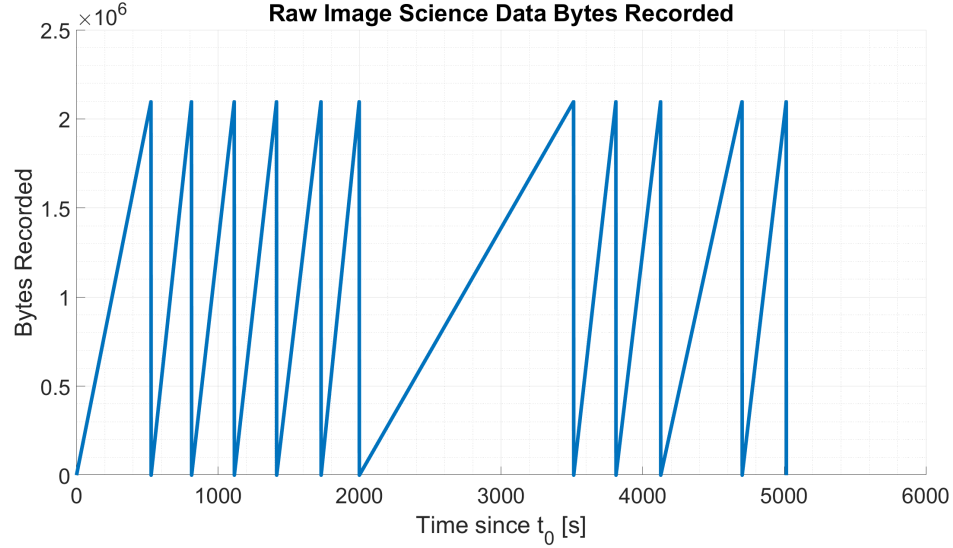
Linear Best Fit	$8.065e-07t + 0.000132$
Mean Std. Error	0.000271
Extrapolated Fill Time [s]	1053048.250 (12.188 days)
Extrapolated Data Volume	0.849
Extrapolated Standard Error	0.0456
Extrapolated 95% Pred. Interval	$0.849 \pm 0.0913$

To assess the performance of the onboard communications system during mission data donwlink operations to the system ground station, a dedicated simulation was



**Figure 4.11: Single Satellite Mission Nadir Angle Time Series**

created to estimate the data downlink volume capabilities. Initial conditions were identical to the single satellite reference mission conditions described in chapter 3, with the exception of beginning with a storage utilization of 80% of onboard science data memory. The onboard mission data storage utilization time series during downlink task execution is shown in Figure 4.15. Note that the trend reflects the default zero-order hold state behavior of the Horizon Simulation Framework, and does not represent the literal linear trend in data storage utilization. The task duration is shown enclosed within the blue dashed line box, and represents a change in onboard memory storage of approximately 3% over the course of a pass approximately five minutes in duration. This is roughly equivalent to 0.6% of onboard memory transmitted per minute, assuming no losses due to packet transmission protocols or a less than ideal duty cycle. Assuming at least two of these passes of no less than five minutes occur over the same single ground station every day, it is estimated that the entire collection of onboard science data could be transmitted in no fewer than 12 days. However, inefficiencies due to weather conditions or atmospheric effects, transient link loss effects, packet transmission protocols, local terrain obscuration

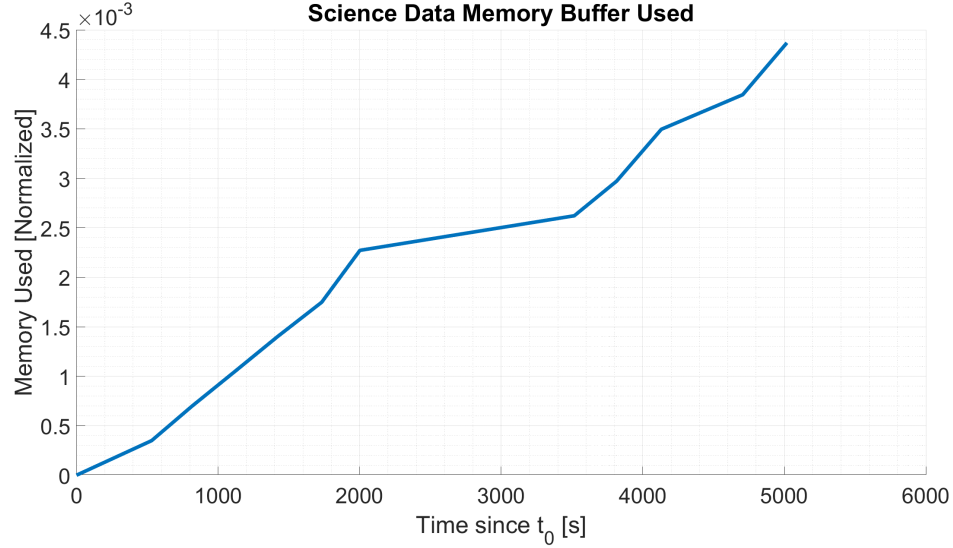


**Figure 4.12: Single Satellite Mission Raw Bytes Recorded Time Series**

effects, or loss or re-establishment of link due to any of the above or other effects are not accounted for in this model. Actual downlink capabilities may prove to be significantly more limited due to any or all of these effects: thus, the relative speed at which science data can be generated may exceed the data throughput capabilities for downlinking this data to ground segment facilities. To obtain a more accurate estimation of the data throughput capabilities of the CubeSat Astronomy Network, more robust and detailed models for telemetry, tracking, and command subsystems, and command and data handling subsystems may need to be developed to provide a higher fidelity characterization of overall system performance.

In the context of the full system architecture, the space-to-ground communications link poses the lowest-bound constraint on the science data throughput capabilities of the system. Unless long-duration, infrequent observation schedules are leveraged, communications with a single ground station may significantly limit the data acquisition and distribution capabilities of the overall system. This disadvantage may not be as severe for comparable ground-based autonomous telescope networks. However,



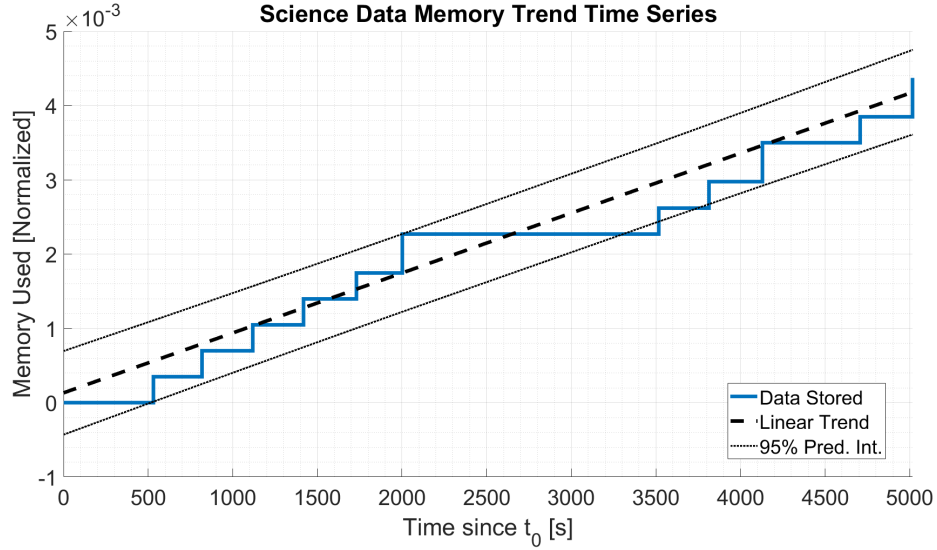


**Figure 4.13: Single Satellite Mission Data Storage Utilized Time Series**

this disadvantage can be addressed to improve system feasibility in one or more of the following ways, though not exclusively to these:

- Data volume reduction via subfield selection and higher-ratio compression methods.
- Modulation schemes with lower signal-to-noise ratio requirements for equivalent bit error rate.
- Utilization of additional high-gain S-band ground stations to provide more frequent mission data downlink opportunities.
- Adoption of novel communications technologies with higher data throughput capabilities, such as free-space optical communication.

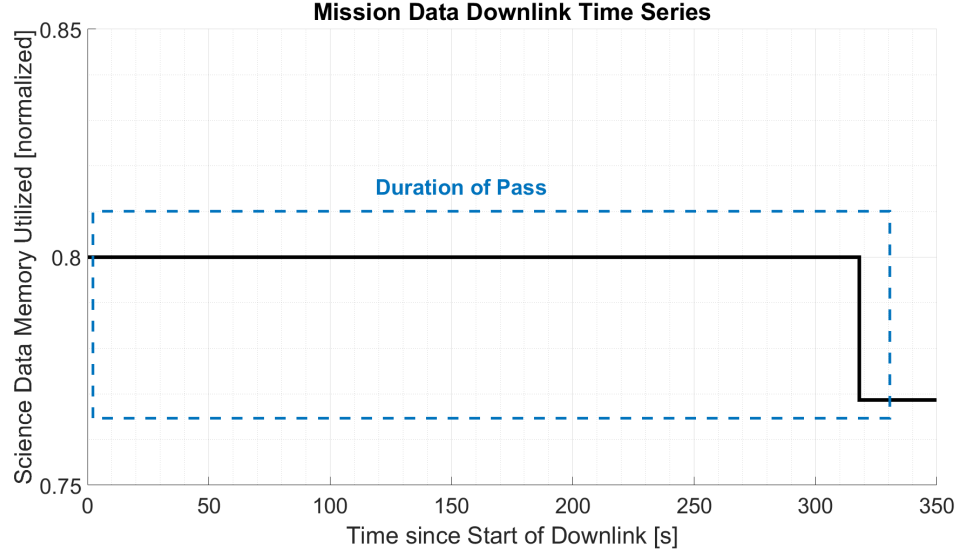
The Optical Communications and Sensors Demonstration mission demonstrated the capabilities of space-to-ground "lasercom" downlink from a CubeSat spacecraft. Per literature by Rose et al., "communication links up to 100 Mbps with bit error rates near  $10^{-6}$  without any forward error correction" [58] were demonstrated in flight



**Figure 4.14: Mission Data Storage Utilized Trend Time Series**

testing during downlink operations from a CubeSat in low earth orbit communicating with a ground station facility in El Segundo, California. Considering that this link was possible with a  $3\sigma$  pointing error of approximately 0.024 degrees, such transmission capabilities may be possible given the required attitude control capabilities of the CubeSat Astronomy Network spacecraft. Such throughput capabilities warrant consideration during future design activities, as such transmission capabilities would lessen the severity of downlink data constraints in the system's present configuration. Adoption and development of such technology, however, may require thorough research of existing regulations, to ensure not only compliance with safety regulations, such as the Federal Aviation Administration's guidelines and regulations on laser operation, but also any future regulation that may be introduced by regulatory agencies, such as the Federal Communications Commission.

The amount of available incident solar power is plotted in Figure 4.16. Note the correlation of the data with Figure 4.9, due to the cosine relationship between angle of incident light and power generated in the model. Due to the power-positive design of the onboard power system, depth of discharge remained negligible throughout the



**Figure 4.15: Downlink Data Storage Trend Time Series**

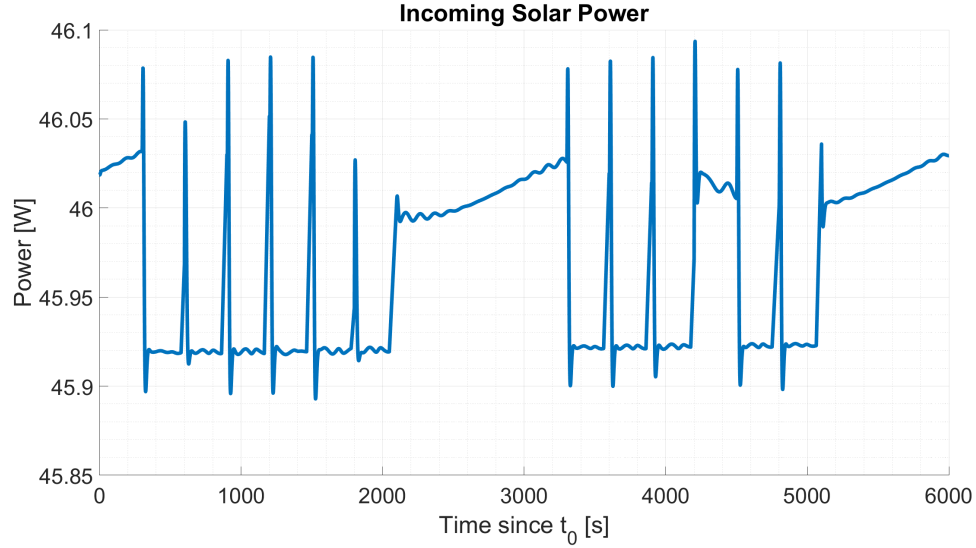
simulation, demonstrating one of the key benefits of the continuously-illuminated sun-synchronous orbit. Further analysis may be required to assess the need for onboard propulsion for orbit maintenance, which may change onboard power and electrical energy storage requirements.

### 4.3 Constellation Case Results

Modeling the constellation reference mission resulted in similar, yet distinct simulation results as shown in the data plots in Appendix A. Differences in state data between assets can chiefly be attributed to differences in initial dynamic state conditions, but also to the ability to satisfy state constraints at different points in each asset's orbit during task execution.

#### 4.3.1 Constellation Attitude Results

The constellation spacecraft attitude time series data for each individual asset is shown in Figure A.1, Figure A.2, and Figure A.3 for Asset 1, Asset 2, and Asset 3, respectively. Note the similarity to the data plotted in Figure 4.2.



**Figure 4.16: Single Satellite Mission Solar Power Input Time Series**

The angular body rates for the constellation satellites are plotted in Figure A.4, Figure A.5, and Figure A.6 for Asset 1, Asset 2, and Asset 3, respectively. Note the similarity to the results shown in Figure 4.4.

Relative angle between satellite boresight axis and the Sun is plotted for each satellite in Figure A.7, Figure A.8, Figure A.9 for Asset 1, Asset 2, and Asset 3, respectively. Note the similarity to the results plotted in Figure 4.9.

Relative angle between satellite boresight axis and the satellite ram direction is plotted for each satellite in Figure A.10, Figure A.11, Figure A.12 for Asset 1, Asset 2, and Asset 3, respectively. Note the similarity to the results plotted in Figure 4.10.

Relative angle between satellite boresight axis and the local nadir direction is plotted for each satellite in Figure A.13, Figure A.14, Figure A.15 for Asset 1, Asset 2, and Asset 3, respectively. Note the similarity to the results plotted in Figure 4.11.

### 4.3.2 Constellation Reaction Wheel Results

Reaction wheel speeds for the individual satellites in the constellation are plotted in Figure A.16, Figure A.17, and Figure A.18, for Asset 1, Asset 2, and Asset 3, respectively. Note the similarity with the data shown in Figure 4.5.

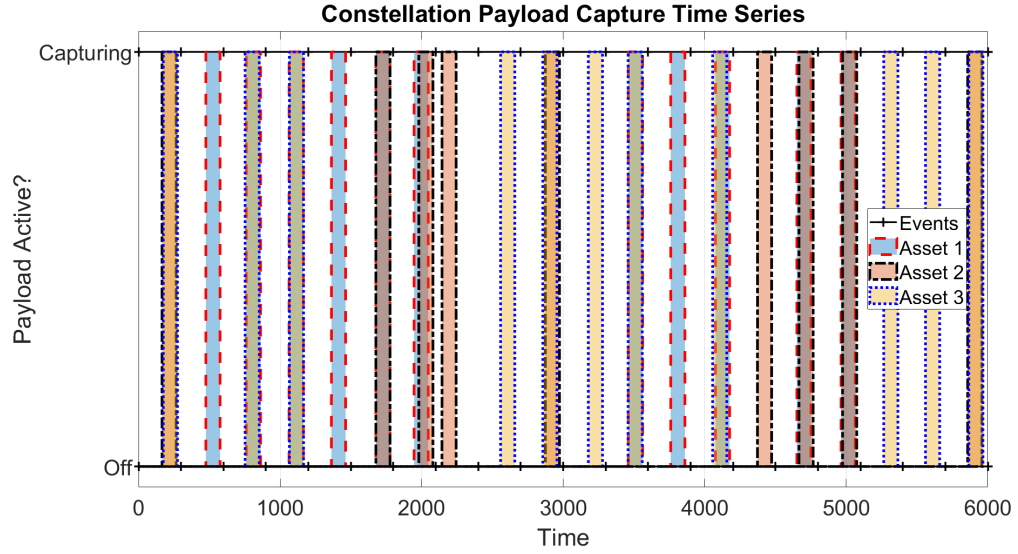
Reaction wheel command torques are plotted for the constellation satellites in Figure A.19, Figure A.19, and Figure A.19 for Asset 1, Asset 2, and Asset 3, respectively. Note the similarity to the results for the single satellite mission shown in Figure 4.8.

### 4.3.3 Constellation Science Data Results

The raw science data recorded during the course of mission operations by each satellite are recorded in Figure A.22, Figure A.23, and Figure A.24 for Asset 1, Asset 2, and Asset 3, respectively. Note the similarity to the results shown in Figure 4.12.

The onboard compressed science data storage utilized by each satellite is plotted in Figure A.25, Figure A.25, and Figure A.25 for Asset 1, Asset 2, and Asset 3, respectively. Note the similar results shown in Figure 4.13.

Constellation target coverage during the simulation timespan, represented as the time when the imaging payload executed as an imaging task, is plotted for all satellites in the constellation in Figure 4.17. Note how no individual satellite can achieve continuous target coverage in every possible event. Additionally, the plotted results demonstrate the execution of an imaging task by at least one satellite during each scheduled event window during the simulation, demonstrating the viability of a constellation-based approach to continuous astronomical observation targets. Thus, an observation schedule with a sample frequency of approximately one observation per five minutes can be achieved continuously during the course of a single orbit by the constellation, if not by an individual spacecraft.



**Figure 4.17: Constellation Payload Capture Time Series**

#### 4.3.4 Constellation Power Generation Results

Onboard solar power generation for each satellite in the constellation are plotted in Figure A.28, Figure A.29, Figure A.30 for Asset 1, Asset 2, and Asset 3, respectively. Note the similarity to the results shown in Figure 4.16.

### 5.1 Summary of Results

The results obtained from the simulated results of modeling the CubeSat Astronomy Network suggest that the constellation mission concept may conditionally satisfy the existing system-level requirements for continuous observation capabilities. Requirements that may still require determination are largely driven by payload and observation requirements, especially pointing stability, target acquisition, and dwell time requirements. The capabilities demonstrated by the CubeSat Astronomy Network spacecraft as modeled in the simulation cases was able to operate over the duration of a single orbit, but longer duration simulations may prove troublesome to model due to the scheduling algorithm's time complexity quadratic relationship with the number of simulation time steps. Thus, either a novel task scheduling algorithm may need to be developed, or a different modeling technique may be required that is presently beyond the current capabilities of Horizon. However, the most important aspect of the system's capabilities, namely the ability to perform uninterrupted time-series observations of a science target using no less than three satellites, was effectively demonstrated over the duration of a single orbit.

Additional capabilities and design considerations, some of which may require the development or maturation of novel technologies, may be required in order to satisfactorily collect and distribute astronomical science data to effectively complement the capabilities of comparable autonomous ground-based telescope networks. These capabilities include, but are not limited to: payload fine-pointing capabilities on the order of sub-arcsecond to arcsecond-scale stability over the course of individual observations, enabled by the use of two-stage attitude control for attitude control and

image stabilization and / or low-noise momentum exchange devices and low-error attitude sensors, per the pointing error results; pre-biased reaction wheel speed commands before observations to minimize the transient attitude error effects of wheel speed reversals; high data bandwidth downlink transmission speeds, on the order of 1-10 megabits per second, or sufficient data reduction to enable the utilization of existing amateur radio ground station hardware to enable data transmission with bandwidths on the order of 1-10 megabits per second; sufficiently high link margins to enable the prior-mentioned data downlink capabilities; sufficiently radiation-hardened components to survive the environment of a sun-synchronous orbit.

## **5.2 Lessons Learned**

Key lessons learned during the course of this thesis pertaining to the usage of the Horizon Simulation Framework for completing model-based systems engineering activities to support the design of the CubeSat Astronomy Network are summarized below. The relevant takeaways from work undertaken during the course of this thesis can be categorically summarized as follows:

- Concurrent development of models and system architecture and requirements may be unavoidable, but can still be managed using sound systems engineering methodologies.
- Model architectures and system architectures, while similar, may serve slightly different functional purposes and thus be configured differently.
- The existing learning curve associated with leveraging the Horizon Framework for model-based systems engineering can be overcome through a combination of sound and efficient modular model design, unit testing, and rigorous full model testing, in conjunction with thorough but efficient documentation and other sound programming practices. This is elaborated upon further in chapter 6.



### 5.2.1 Concurrent Development Practices

One of the key benefits of model based systems engineering methodologies is the capacity for reuse of toolsets developed for modeling and simulation of similar systems. Similarly to how software packages and modules can be reused for future software development, analytical toolsets used in the process of requirements verification of a particular system can be reused and adapted for future similar systems. Additionally, minimizing the time required to implement and test system changes using prior developed models can permit greater levels of automated task management and workflow into the required processes of system design and analysis, which was leveraged during the course of this thesis work by utilization of a centralized remote repository hosted by GitHub to store and provide version control for the latest version of the Framework and any new code that was required for modeling the Network.

One key discovery during this process was that uncertainty in the results of tradeoffs associated with the design of the network translated into uncertainty in the design of the models that were required to simulate the performance of the system: hence, certain arbitrary assumptions had to be made at stages in the development process. These decisions were used to test the novel capabilities of more robust features that would be able to be further developed to enable parametric trade studies that would either validate or invalidate these originally arbitrary design decisions, and would allow the models to mature and develop in tandem with the design of the Network. The necessary final state of these models would enable the requisite tradeoffs to be validated with the mission concept, ideally in the form of a fully parameterized simulation of the mission over the duration of the Network's operational lifetime. However, the simulated performance of ongoing nominal mission operations with the current state of system requirements was deemed suitable for mission concept validation at this stage of design, due to the potentially unbounded lifetime capabilities of the sys-

tem as a whole (enabled by the potentially low unit cost of assets for constellation replenishment).



### 5.2.2 Model and System Architectures: Similarities and Differences

Despite the parallel development capabilities for systems and their modeled representations, a lack of proper distinction between their architectures can pose troublesome implications for model development. While the primary intent of a system is, generally speaking, to satisfy its requirements, the primary intent of a model is to represent the characteristics and behavior of a system to a required degree of fidelity for a required purpose. Thus, the effectiveness of a model is not only characterized by its accuracy, but its effectiveness in accomplishing its required purpose.

Prior work accomplished with the Framework included: characterizing the on-orbit performance of the hypothetical Aeolus constellation; modeling and validating the flight performance, stability, and control systems of sounding rockets; and assessing the feasibility of utilizing pheromone-based path planning for unmanned aerial systems. To date, it is unknown if any prior applications of the Framework have included the validation of system requirements for a proposed constellation of deep-space observation satellites, though the usage of non-linear attitude dynamic models for requirements validation for such a system has been demonstrated in the presented results. However, usage of the Framework explicitly as a design tool, applied in as readily a fashion as a systems engineer might use a commercial analogue such as AGI's Systems ToolKit or NASA's open-source General Mission Analysis Tool, is apparently unprecedented in the Framework's development history. Thus, this apparently novel application of Horizon had little development heritage to draw upon for the purposes of model development, though thorough documentation on the Framework and its prior applications by Yost [12], Mehiel [37], MacLean [31], and Frye [40] proved useful during the process of model development.

The chief difficulty encountered during model development came from answering as the designer the following question: "How can model dynamics and the subsystems that depend on dynamic state information be modeled deterministically when the scheduled execution of tasks is modeled non-deterministically?" In other words, how can dynamic state and system state be propagated together with mutual interdependencies on one another? In essence, the execution of certain subsystem methods during an event for the CubeSat Astronomy Network requires prior knowledge of dynamic state data during the simulated event. Equations of motion, as currently defined in Horizon, parametrically describe the state transition relationships for an asset's dynamic state, and parameters external to the equations of motion, such as controller gain values or target positions, can be input to the equations of motion to model the resulting dynamic performance. Analogously to how subsystems can be dependent on state information from one another, assets' equations of motion can be dependent on asset subsystems to input target, environment, or subsystem state information in order to properly model the changes in assets' dynamic states.

For the Aeolus mission test case, the equations of motion were time-invariant and deterministic, with no explicit dependencies on subsystem information. In the case of MacLean's sounding rocket models, while the state machine logic leveraged by the subsystem models was deterministic, the actual equations of motion required system state information in order to propagate the asset's dynamic state. This was accomplished by running the scheduler at a rate equivalent to the comparable rate of a sounding rocket flight computer, which would check at a rate of 50 Hz if the rocket needed to deploy a drogue chute or a recovery parachute, or continue controlled flight by means of tail fin deflections. Aerodynamic coefficients that would change based on chute deployment or fin deflection were input as parameters to the equations of motion by their responsible subsystems, enabling system state information to be used to propagate the rocket's dynamic state. However, this meant that the system

was required to successfully execute a task at every simulation time step, since state information cannot propagate and a task cannot be successfully scheduled unless every subsystem during an event can perform the task while satisfying all system constraints.

### 6.1 Summary of Simulated Results

In summary, the simulated results suggest that per chapter 4 and chapter 5, at least over the time period of single to multiple orbits, the proposed concept of operations for achieving continuous science target coverage using a low earth orbit constellation of space telescopes is not infeasible. However, further design work and model development is necessary to assess the feasibility of the mission concept beyond what has been demonstrated within the scope of this thesis. The necessary steps required for both expanding the capabilities of Horizon to accomplish this characterization, as well as the Network’s design itself, are describe in further detail below.

### 6.2 Summary of Design Assessment

To summarize per section 5.1, the continuous observation of astronomical science targets using a constellation of spacecraft in low earth orbit may be possible with no more than three equidistant spacecraft in a single orbital plane. The key enabling design factors for the CubeSat Astronomy Network architecture in its present configuration include, but are not limited to: high-precision, long-duration pointing capabilities, dependent on the exact stability and scientific precision requirements of the instrument, likely with the addition of image-stabilization capability within the spacecraft payload itself similarly to the ASTERIA imaging payload[30]<sup>1</sup> with the ability to maintain precise pointing of the spacecraft bus within 1-10 arcseconds (dependent on the instruments pixel scale and the required signal-to-noise ratio) of a target for the

---

<sup>1</sup>ASTERIA’s payload leveraged an onboard Fairchild CIS2521 frontside illuminated CMOS sensor[30] with a piezoelectrically-actuated imaging platform precisely for the purpose of enabling sub-arcsecond image stabilization during scientific observations.

duration of an image capture; low noise, high accuracy attitude sensors and actuators with a well-tuned and robust attitude control architecture to enable such fine pointing, likely possible with LQR setpoint control with specifically-tuned pointing modes; and high bandwidth science data downlink capability, to maximize the time spent conducting observations on-orbit, possibly enabled using an upgraded ground station at Cal Poly’s existing PolySat ground station[41] or leveraging recent developments in CubeSat-based optical communication to a capable ground station[58].

### **6.3 Summary of Necessary Design Changes and Analysis**

Key design changes and requirements to be confirmed that are necessary for development of the CubeSat Astronomy Network mission concept include, but are not limited to: investigation and assessment of the feasibility of implementing fine-pointing attitude stabilization in the onboard science imaging payload, possibly using a similar approach leveraged by the ASTERIA mission; derivation of more complete science data storage requirements, including requirements on data compression, sub-field image processing, and other data reduction methods to maximize the scientific value of downlinked mission data products by minimizing the transmission of data that does not improve the fidelity of observation results; and comprehensive analysis of thermal variations on system performance, due to the lack of thermal control system modeling within the scope of the work performed in this Thesis.

### **6.4 Summary of Assessment of Horizon Simulation Framework**

As has been demonstrated within the scope of this thesis, custom model were developed and tested to model the performance of system activities by the CubeSat Astronomy Network, demonstrating the capability of the Framework to model the performance of CubeSat space systems. As is explained further in subsection 6.5.1, extension of the Framework’s capabilities to interact with a wider range of engineering

and design software will enable the Framework to capitalize on existing space mission development tools to provide a broader range of functionality to users of the Framework. However, such extension should be performed with unit testing to assess the impact on simulation performance, if any such adverse runtime performance effects are possible.

As was discussed in subsection 5.2.2, the architectural distinction between a subsystem state and an asset’s dynamic state in the Horizon Simulation Framework posed a challenge when attempting to model interdependencies between the two. Theoretically, the modeling of a subsystem to propagate relevant dynamic state information for individual subsystems, in essence modeling an asset’s ”knowledge” of its own dynamic state, may be sufficient to enable the utilization of an asset’s dynamic state information to propagate subsystem state information. Such a subsystem’s capability would make it analogous to a navigation subsystem, and the performance of representative components of such a navigation subsystem have been modeled successfully in MacLean’s work[31]. Alternatively, a refactoring of the architecture may enable a more consistent application of subsystem states and asset dynamic states, since the asset dynamic state can be viewed as an analog of subsystem dynamic state that is representative of the associated subsystems of a given asset.

## **6.5 Future Work**

Throughout the process of development using the Horizon Simulation Framework, several opportunities for further development of Horizon’s capabilities were identified as key opportunities to expand the existing modeling and simulation capabilities of the framework. Additionally, future design work in maturing the CubeSat Astronomy Network that can be informed by the novel capabilities of Horizon that were implemented as a result of this thesis work was identified and is summarized below. The relevant design implications for the CubeSat Astronomy Network as a result of

this thesis and how those relate to future design tasks are also presented. Finally, the immediate next steps in the design of the CubeSat Astronomy Network are outlined and presented below, given the results of this thesis' results.

#### **6.5.1 Interoperability with CubeSat Reference Model**

Due to the potential robustness and reusability of existing and planned modeling standards for CubeSat missions, including the CubeSat Reference Model [18], it is recommended that the functionality to enable interfacing with models developed for CubeSat mission modeling be integrated into the Horizon Simulation Framework. Horizon model input file generation from SysML diagrams was previously as part of Luther's thesis work [10], and part of this thesis' work involved the expansion of parametric relationship and functional performance models for CubeSat spacecraft: therefore, continued expansion of modeling capabilities for CubeSat missions using Horizon and interoperability with existing model-based systems engineering toolsets seem logical areas for future Horizon development work.

#### **6.5.2 Interoperability with Design and Analysis Software**

Prior work by Frye [40] and MacLean [31] demonstrated the effectiveness of external aerodynamic modeling software for generating linearized flight dynamic models for simulation atmospheric craft performance. Work accomplished during the course of this thesis demonstrated how existing APIs could be leveraged to enable direct interoperability between external software and file types and models utilized by the Horizon Simulation Framework. Further development of direct interoperability with software external to the Framework could expand the modeling capabilities of Horizon, and improve the model development time required if certain external toolsets are necessary for determining parameters already leveraged by Horizon models. Examples of particularly valuable candidates for development of interoperability capabilities



include: Mathworks' MATLAB, which is highly integrated into the Aerospace Engineering curriculum at Cal Poly; Daussalt Systemes' Solidworks, which had already been leveraged for design work for the CubeSat Astronomy Network that transpired prior to the work accomplished for this thesis; the Massachusetts Institute of Technology's AVL software, a computational fluid dynamics program specialized for using vortex lattice method modeling to determine flight characteristics for aircraft models, which found use previously in Frye's and MacLean's respective theses; and Analytical Graphics, Inc.'s Systems Tool Kit, a simulation software commonly leveraged for CubeSat mission design.

### **6.5.3 Additional Functionality in Horizon**

While a tremendous amount of functionality currently exists within the framework, there are some software development practices that could pose tremendous benefits for leveraging Horizon for future system design activities. Auto-generated documentation, using the existing Horizon XML documentation comments (in addition to additional comments for undocumented existing code) can be created using a variety of free and open-source software, and then be hosted on the official Horizon repository on GitHub. New class diagrams for each assembly in the Horizon source code have also been generated and are accessible in Visual Studio from each assembly.

NuGet packages that are hosted online may also be leveraged to enable even greater functionality for Horizon models, especially those with direct applicability to expanding the existing mathematical modeling functions, such as the Math.NET project and its numerical computing and signal processing toolkits. While care must be taken to manage the dependencies of external software packages and prevent conflicts with the existing Horizon architecture, the communities associated with these open source frameworks can prove valuable during the course of integration into Horizon and debugging models during development.

Furthermore, some aspects of the architecture of Horizon remain at a particular level of functionality either due to design decisions made during previous versions that can restrict modeling options for a developer. While many facets of the core architecture of the Framework remain watertight and aligned with sound object-oriented programming practice, new developers that may have less experience with object-oriented development may find these decisions more arbitrary and unintuitive than they may actually be. A simple walkthrough or tutorial demonstrating the features of the Horizon Simulation Framework, even simpler than the Aeolus demonstration mission, may sufficiently decrease the necessary training time for a new developer to comfortably begin leveraging the Framework for their own model development and testing, though this was not in the scope of this thesis.

#### **6.5.4 Next Steps in CubeSat Astronomy Network System Design**

Due to the promising results of this analysis, it is anticipated that more detailed design of the CubeSat Astronomy Network to the level of detail required for a Preliminary Design Review should be completed. This will require more detailed spacecraft design, especially at the subsystem level. However, before such preliminary design activities commence, a complete and thorough system requirements review should be conducted to ensure that design decisions and tradeoffs are performed with as complete an understanding of the required system parameters. Especially pertinent to completing the collection of system level requirements will be the more detailed design of a science payload or range of science payloads suitable for use in the CubeSat Astronomy Network.

## BIBLIOGRAPHY

- [1] J. A. Estefan *et al.*, “Survey of model-based systems engineering (mbse) methodologies,” *Incose MBSE Focus Group*, vol. 25, no. 8, pp. 1–12, 2007.
- [2] W. Weiss, S. Rucinski, A. Moffat, A. Schwarzenberg-Czerny, O. Koudelka, C. Grant, R. Zee, R. Kuschnig, J. Matthews, P. Orleanski, *et al.*, “Brite-constellation: nanosatellites for precision photometry of bright stars,” *Publications of the Astronomical Society of the Pacific*, vol. 126, no. 940, p. 573, 2014.
- [3] C. M. Pong, “On-orbit performance and operation of the attitude and pointing control subsystems on asteria,” in *Proc. of the 32nd Annual AIAA/USU Conf. on Small Satellites*, 2018.
- [4] A. Johnson, C. Van Steenwyk, R. Genet, and D. Rowe, “Cubesat astronomy network,” 2019.
- [5] A. Feldman, K. A. Divoll, and A. Rogan-Klyve, “Becoming researchers: The participation of undergraduate and graduate students in scientific research groups,” *Science Education*, vol. 97, no. 2, pp. 218–243, 2012.
- [6] M. T. Fitzgerald, R. Hollow, L. M. Rebull, L. Danaia, and D. H. McKinnon, “A review of high school level astronomy student research projects over the last two decades,” *Publications of the Astronomical Society of Australia*, vol. 31, 2014.
- [7] J. Johnson, R. Genet, J. Armstrong, G. Boyce, P. Boyce, M. Brewer, R. Buchheim, J. Carro, R. Estrada, C. Estrada, R. Freed, S. Gillette, R. Harshaw, T. Hollis, J. Kenney, S. McGaughey, C. McNab, K. Mohanan, B. Sepulveda, D. Wallace, and V. Wallen, “Astronomy Research Seminar,”

*Society for Astronomical Sciences Annual Symposium*, vol. 34, pp. 139–148, May 2015.

- [8] R. M. Genet, J. M. Johnson, and V. Wallen, “One-semester astronomical research seminars,” *Small Telescopes and Astronomical Research*, 2010.
- [9] M. Hause *et al.*, “The sysml modelling language,” in *Fifteenth European Systems Engineering Conference*, vol. 9, pp. 1–12, Citeseer, 2006.
- [10] S. Luther, “Sysml based cubesat model design and integration with the horizon simulation framework,” Master’s thesis, California Polytechnic State University - San Luis Obispo, Jul 2016.
- [11] V. K. Patel, “Sysml output interface and system-level requirement analyzer for the horizon simulation framework,” Master’s thesis, California Polytechnic State University - San Luis Obispo, Jul 2018.
- [12] M. Yost, “An iteration on the horizon simulation framework to include .net and python scripting,” Master’s thesis, California Polytechnic State University - San Luis Obispo, Aug 2016.
- [13] I. Lunsford, “Schedule failure analysis within the horizon simulation framework,” Master’s thesis, California Polytechnic State University - San Luis Obispo, Jul 2016.
- [14] R. Karban, F. G. Dekens, S. Herzig, M. Elaasar, and N. Jankevičius, “Creating system engineering products with executable models in a model-based engineering environment,” in *Modeling, Systems Engineering, and Project Management for Astronomy VI*, vol. 9911, p. 99110B, International Society for Optics and Photonics, 2016.

- [15] A. Mehrparvar, D. Pignatelli, J. Carnahan, R. Munakat, W. Lan, A. Toorian, A. Hutputanasin, and S. Lee, “Cubesat design specification rev. 13,” *The CubeSat Program, Cal Poly San Luis Obispo, US*, vol. 1, p. 2, 2014.
- [16] S. C. Spangelo, D. Kaslow, C. Delp, B. Cole, L. Anderson, E. Fosse, B. S. Gilbert, L. Hartman, T. Kahn, and J. Cutler, “Applying model based systems engineering (mbse) to a standard cubesat,” in *Aerospace Conference, 2012 IEEE*, pp. 1–20, IEEE, 2012.
- [17] D. Kaslow, L. Anderson, S. Asundi, B. Ayres, C. Iwata, B. Shiotani, and R. Thompson, “Developing and distributing a cubesat model-based system engineering (mbse) reference model,” in *Proceedings of the 31st Space Symposium*, pp. 1–14, 2015.
- [18] D. Kaslow, B. Ayres, P. T. Cahill, L. Hart, A. G. Levi, and C. Croney, “Developing an mbse cubesat reference model—interim status# 4,” in *2018 AIAA SPACE and Astronautics Forum and Exposition*, p. 5328, 2018.
- [19] J. Guo, E. Gill, and S. Figari, “Model-based systems engineering to support the development of nano-satellites,” in *Proc. of the 65th International Astronautical Congress*, vol. 10, pp. 6991–7003, 10 2014.
- [20] D. L. Ciperia, “Comparison of traditional versus cubesat remote sensing: A model-based systems engineering approach,” Master’s thesis, Air Force Institute of Technology, 2018.
- [21] M. Waseem and M. U. Sadiq, “Application of model-based systems engineering in small satellite conceptual design-a sysml approach,” *IEEE Aerospace and Electronic Systems Magazine*, vol. 33, no. 4, pp. 24–34, 2018.
- [22] J. Crane and L. Brownlow, “Optimization of multi-satellite systems using integrated model based system engineering (mbse) techniques,” in *Systems*

- Conference (SysCon), 2015 9th Annual IEEE International*, pp. 206–211, IEEE, 2015.
- [23] A. C. Snow, J. L. Worthy III, A. den Boer, L. J. Alexander, M. J. Holzinger, and D. Spencer, “Optimization of cubesat constellations for uncued electrooptical space object detection and tracking,” *Journal of Spacecraft and Rockets*, pp. 401–419, 2016.
- [24] L. Anderson, B. Cole, R. Yntema, M. Bajaj, S. Spangelo, D. Kaslow, C. Lowe, E. Sudano, M. Boghosian, R. Reil, *et al.*, “Enterprise modeling for cubesats,” in *Aerospace Conference, 2014 IEEE*, pp. 1–16, IEEE, 2014.
- [25] N. Schwartz, D. Pearson, S. Todd, M. Milanova, W. Brzozowski, A. Vick, D. Lunney, D. MacLeod, S. Greenland, J.-F. Sauvage, *et al.*, “Laboratory demonstration of an active optics system for high-resolution deployable cubesat,” *arXiv preprint arXiv:1809.09097*, 2018.
- [26] A. Naseri, R. Pirayesh, R. K. Adcock, S. J. Stochaj, N. Shah, and J. Krizmanic, “Formation flying of a two-cubesat virtual telescope in a highly elliptical orbit,” in *2018 SpaceOps Conference*, p. 2633, 2018.
- [27] D. Selva and D. Krejci, “A survey and assessment of the capabilities of cubesats for earth observation,” *Acta Astronautica*, vol. 74, pp. 50–68, 2012.
- [28] G. Handler, A. Pigulski, W. Weiss, A. Moffat, R. Kuschnig, G. Wade, G. Orleański, S. Ruciński, O. Koudelka, R. Smolec, *et al.*, “The brite-constellation nanosatellite space mission and its first scientific results,” in *EPJ Web of Conferences*, vol. 160, p. 01001, EDP Sciences, 2017.
- [29] K. Carroll, S. Rucinski, and R. Zee, “Arc-minute nanosatellite attitude control: Enabling technology for the brite stellar photometry mission,” in *Proc. of the 18th Annual AIAA/USU Conference on Small Satellites*, 2004.

- [30] M. W. Smith, A. Donner, M. Knapp, C. Pong, C. Smith, J. Luu, P. Di Pasquale, R. L. Bocchino Jr, B. Campuzano, J. Loveland, *et al.*, “On-orbit results and lessons learned from the asteria space telescope mission,” in *Proceedings of the AIAA/USU Conference on Small Satellites, The Year in Review, SSC18-I-08*. <http://digitalcommons.usu.edu/smallsat/2018/all2018/08>, 2018.
- [31] S. M. MacLean, “Modeling and simulation of a sounding rocket active stabilization system,” Master’s thesis, California Polytechnic State University - San Luis Obispo, 2017.
- [32] E. L. Shkolnik, “On the verge of an astronomy cubesat revolution,” *Nature Astronomy*, vol. 2, no. 5, p. 374, 2018.
- [33] T. Brown, N. Baliber, F. Bianco, M. Bowman, B. Burleson, P. Conway, M. Crellin, É. Depagne, J. De Vera, B. Dilday, *et al.*, “Las cumbres observatory global telescope network,” *Publications of the Astronomical Society of the Pacific*, vol. 125, no. 931, p. 1031, 2013.
- [34] S. Lampoudi and E. Saunders, “Telescope network scheduling-rationale and formalisms,” *ICORES*, vol. 13, pp. 313–317, 2013.
- [35] M. R. Reddy, “Effect of low earth orbit atomic oxygen on spacecraft materials,” *Journal of Materials Science*, vol. 30, no. 2, pp. 281–307, 1995.
- [36] I. S. in Space, “Magnetorquer board (imtq),” 2016.
- [37] C. O’Connor, E. Mehiel, and B. Butler, “Horizon 2.1: A space system simulation framework,” in *AIAA Modeling and Simulation Technologies Conference and Exhibit*, p. 6546, 2008.

- [38] K. S. Moghaddam and J. S. Usher, “Preventive maintenance and replacement scheduling for repairable and maintainable systems using dynamic programming,” *Computers & Industrial Engineering*, vol. 60, no. 4, pp. 654–665, 2011.
- [39] C. O’Connor, “Horizon: A system modeling and simulation framework for systems engineering utility analysis,” Master’s thesis, California Polytechnic State University - San Luis Obispo, June 2007.
- [40] A. J. Frye and E. A. Mehiel, “Modeling and simulation of vehicle performance in a uav swarm using horizon simulation framework,” in *AIAA Scitech 2019 Forum*, p. 1980, 2019.
- [41] I. M. Bland, “Receive sensitivity characterization of the polysat satellite communication system,” *Master’s Theses and Project Reports*, p. 265, 2010.
- [42] A. H. De Ruiter, C. Damaren, and J. R. Forbes, *Spacecraft dynamics and control: an introduction*. John Wiley & Sons, 2012.
- [43] R. Hevner, W. Holemans, J. Puig-Suari, and R. Twiggs, “An advanced standard for cubesats,” 2011.
- [44] P. C. Hughes, *Spacecraft Attitude Dynamics*. Courier Corporation, 2004.
- [45] D. A. Vallado and W. D. McClain, *Fundamentals of astrodynamics and applications*. Published by Microcosm Press, 4th ed., 2013.
- [46] S. Maus, S. Macmillan, S. McLean, B. Hamilton, A. Thomson, M. Nair, and C. Rollins, “The us/uk world magnetic model for 2010-2015,” 2010.
- [47] B. Hofmann-Wellenhof and H. Moritz, *Physical geodesy*. Springer Science & Business Media, 2006.



- [48] B. C. Technologies, “Attitude control systems,” 2019.
- [49] B. C. Technologies, “Reaction wheels,” 2019.
- [50] L. Sahar, E. Edlerman, H. Agalarian, V. Balabanov, and P. Gurfil, “Hard disk drive based reaction wheels for cubesat attitude control,” *Journal of Spacecraft and Rockets*, vol. 55, no. 1, pp. 236–241, 2017.
- [51] D. Hegel, “Flexcore: Low-cost attitude determination and control enabling high-performance small spacecraft,” 2016.
- [52] J. P. Hespanha, “Lqg/lqr controller design,” *Undergraduate Lecture Notes, University of California, Santa Barbara, California, USA*, 2007.
- [53] J.-F. Tréguët, D. Arzelier, D. Peaucelle, C. Pittet, and L. Zaccarian, “Reaction wheels desaturation using magnetorquers and static input allocation,” *IEEE Transactions on Control Systems Technology*, vol. 23, no. 2, pp. 525–539, 2014.
- [54] W. Pence, R. Seaman, and R. White, “Lossless astronomical image compression and the effects of noise,” *Publications of the Astronomical Society of the Pacific*, vol. 121, no. 878, p. 414, 2009.
- [55] J. A. King, “Amsat/iaru annotated link model system, ver. 2.5.5,” 2016.
- [56] G. Walker, J. Matthews, R. Kuschnig, R. Johnson, S. Rucinski, J. Pazder, G. Burley, A. Walker, K. Skaret, R. Zee, *et al.*, “The most asteroseismology mission: Ultraprecise photometry from space,” *Publications of the Astronomical Society of the Pacific*, vol. 115, no. 811, p. 1023, 2003.
- [57] C. Beierle, A. Norton, B. Macintosh, and S. D’Amico, “Two-stage attitude control for direct imaging of exoplanets with a cubesat telescope,” in *Space Telescopes and Instrumentation 2018: Optical, Infrared, and Millimeter*

- Wave*, vol. 10698, p. 106981Z, International Society for Optics and Photonics, 2018.
- [58] T. Rose, D. Rowen, S. LaLumondiere, N. Werner, R. Linares, A. Faler, J. Wicker, C. Coffman, G. Maul, D. Chien, *et al.*, “Optical communications downlink from a 1.5 u cubesat: Ocsd program,” in *International Conference on Space OpticsICSO 2018*, vol. 11180, p. 111800J, International Society for Optics and Photonics, 2019.
- [59] A. Chulliat, W. Brown, P. Alken, S. Macmillan, M. Nair, C. Beggan, A. Woods, B. Hamilton, B. Meyer, and R. Redmon, “Out-of-cycle update of the us/uk world magnetic model for 2015-2020: Technical note,” 2019.
- [60] D. Plus, “4 mpx cmos camera,” 2018.
- [61] E. Canuto, C. Novara, D. Carlucci, C. P. Montenegro, and L. Massotti, *Spacecraft Dynamics and Control: The Embedded Model Control Approach*. Butterworth-Heinemann, 2018.

## APPENDICES

### Appendix A

#### CONSTELLATION STATE DATA

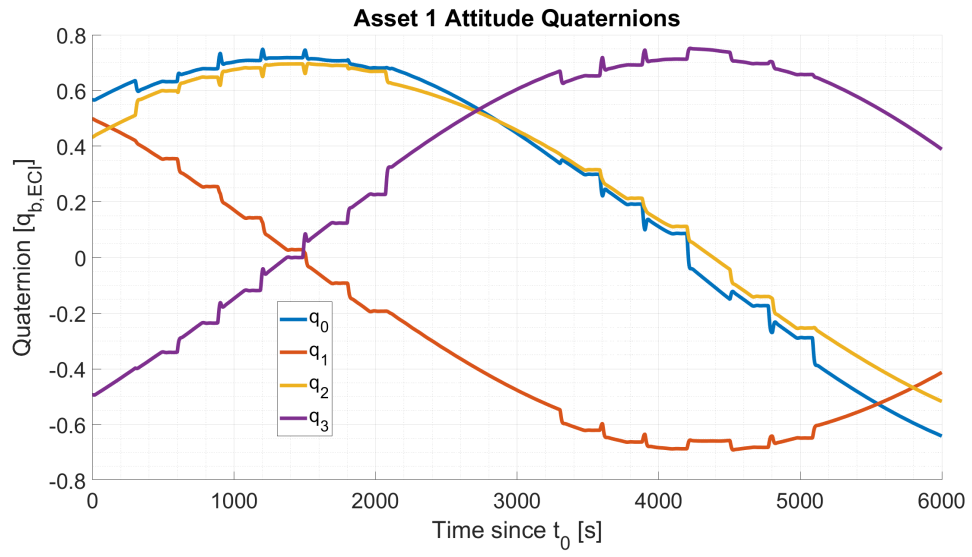


Figure A.1: Constellation Attitude Time Series for Satellite 1

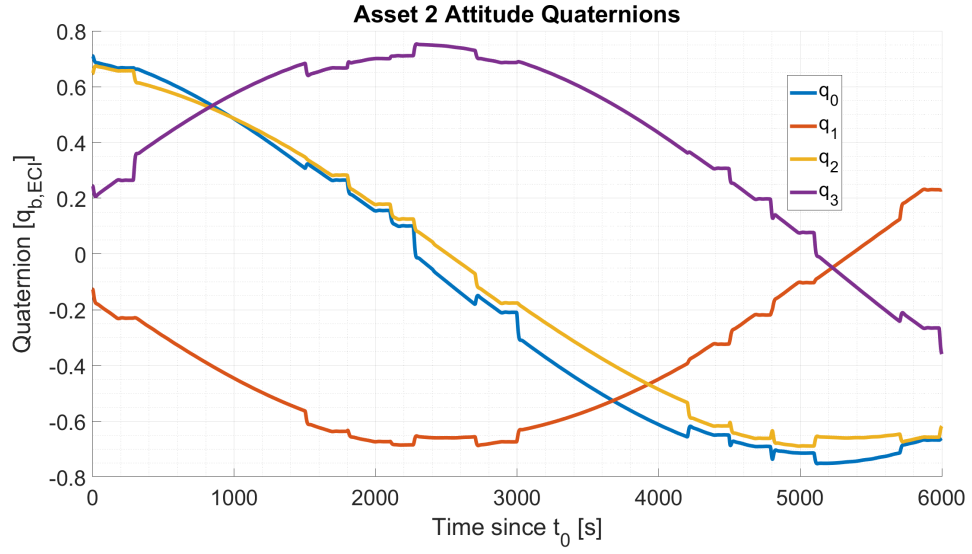


Figure A.2: Constellation Attitude Time Series for Satellite 2

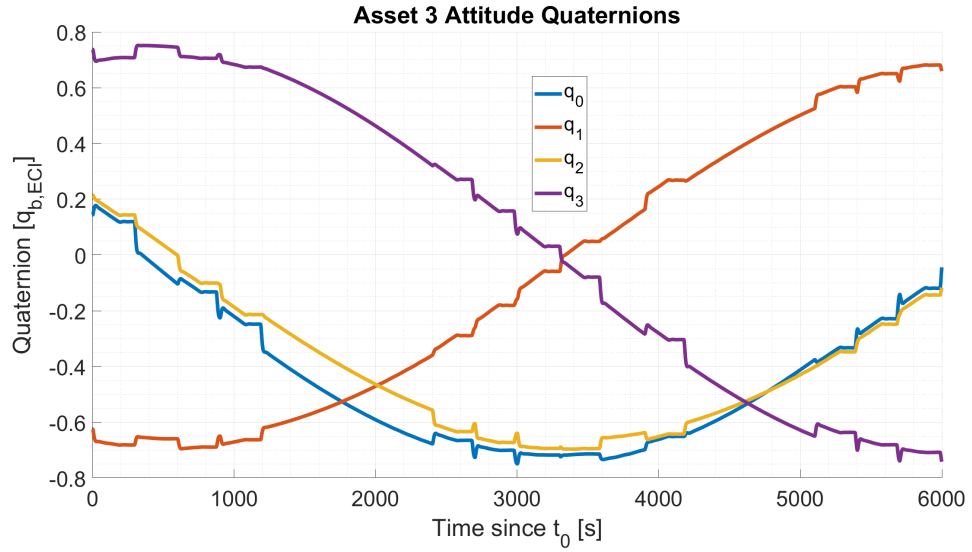


Figure A.3: Constellation Attitude Time Series for Satellite 3

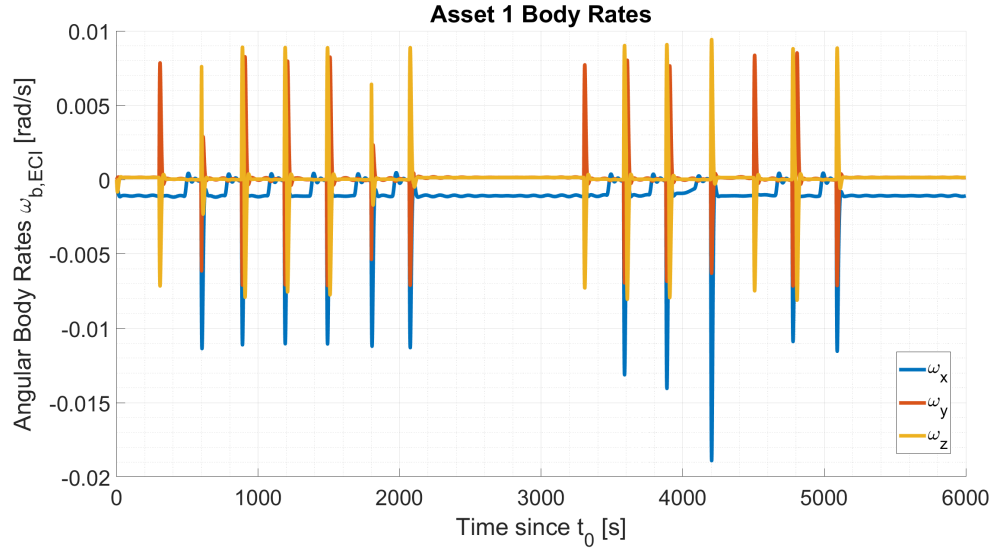


Figure A.4: Constellation Body Rate Time Series for Satellite 1

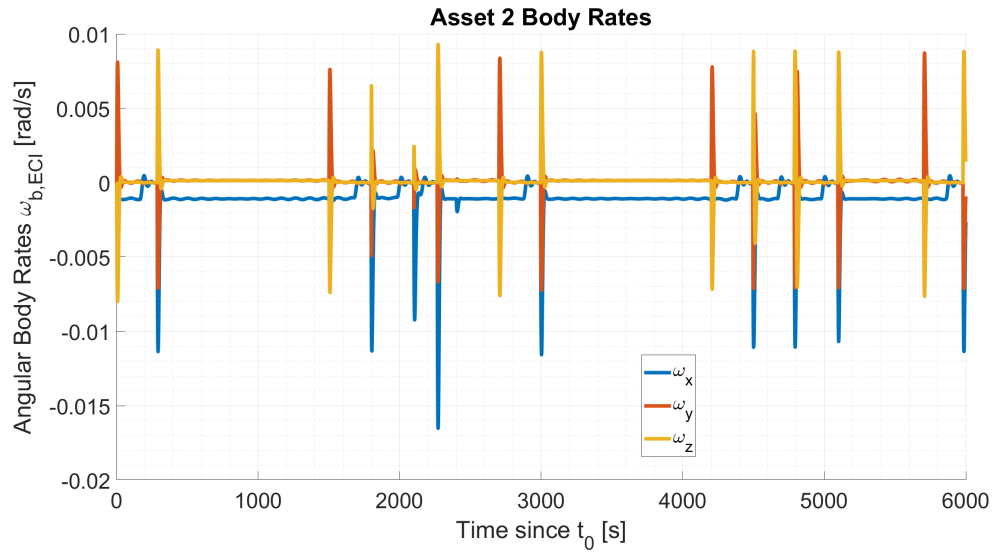


Figure A.5: Constellation Body Rate Time Series for Satellite 2

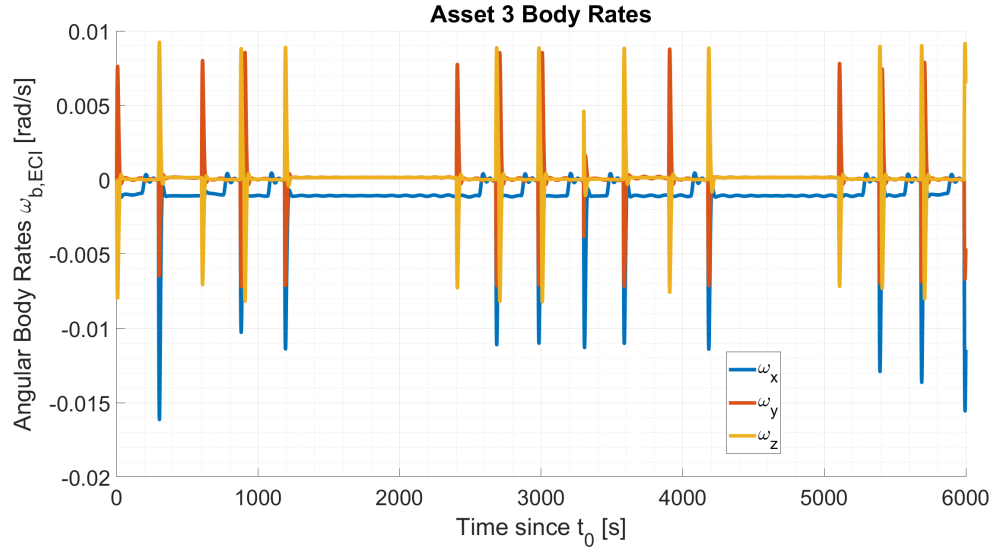


Figure A.6: Constellation Body Rate Time Series for Satellite 3

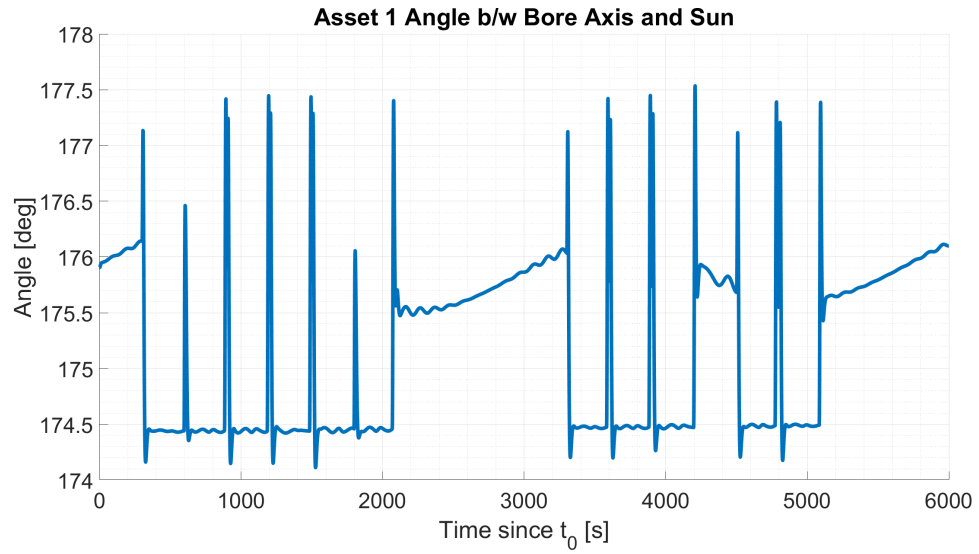


Figure A.7: Constellation Sun Angle Time Series for Satellite 1

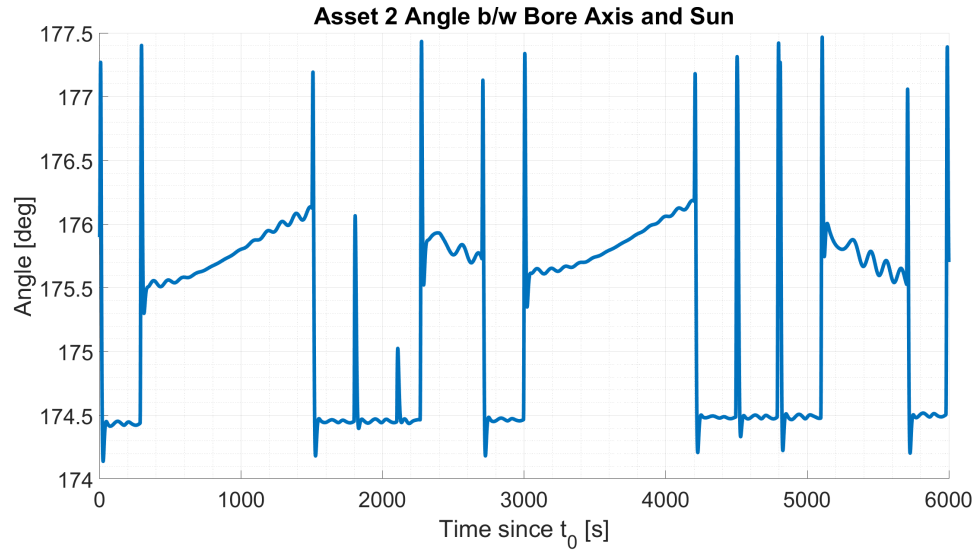


Figure A.8: Constellation Sun Angle Time Series for Satellite 2

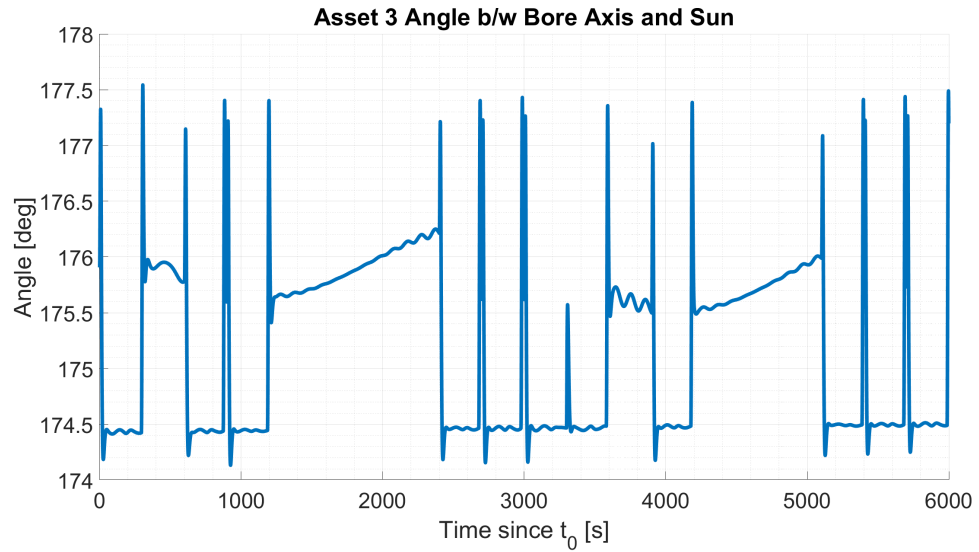
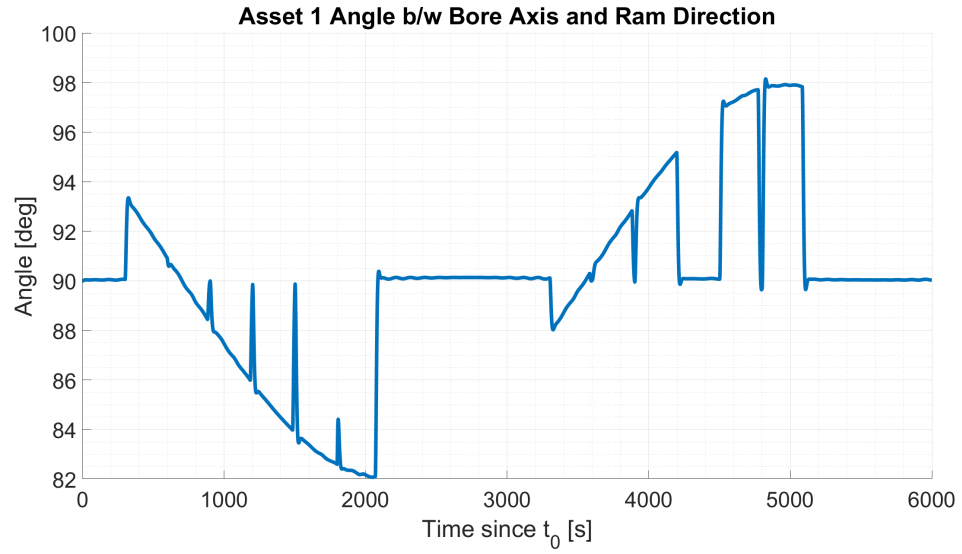
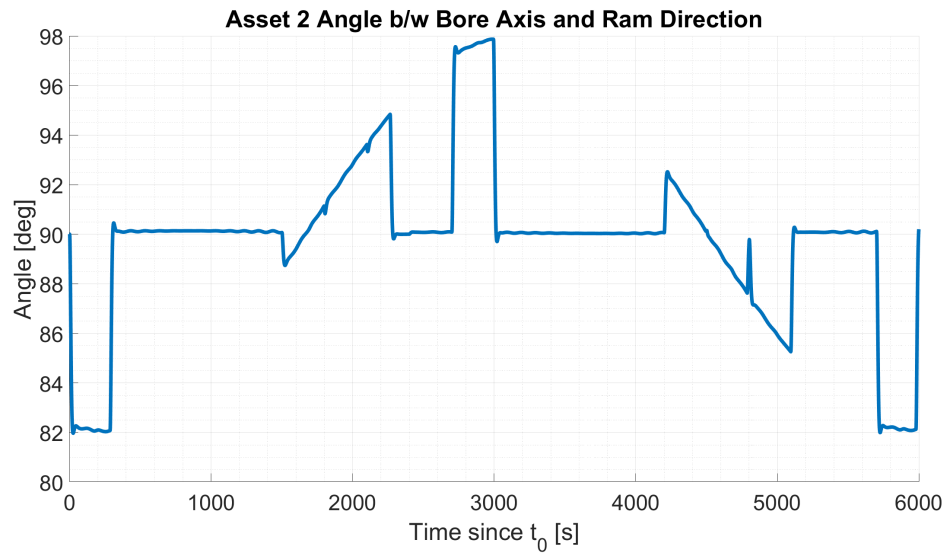


Figure A.9: Constellation Sun Angle Time Series for Satellite 3

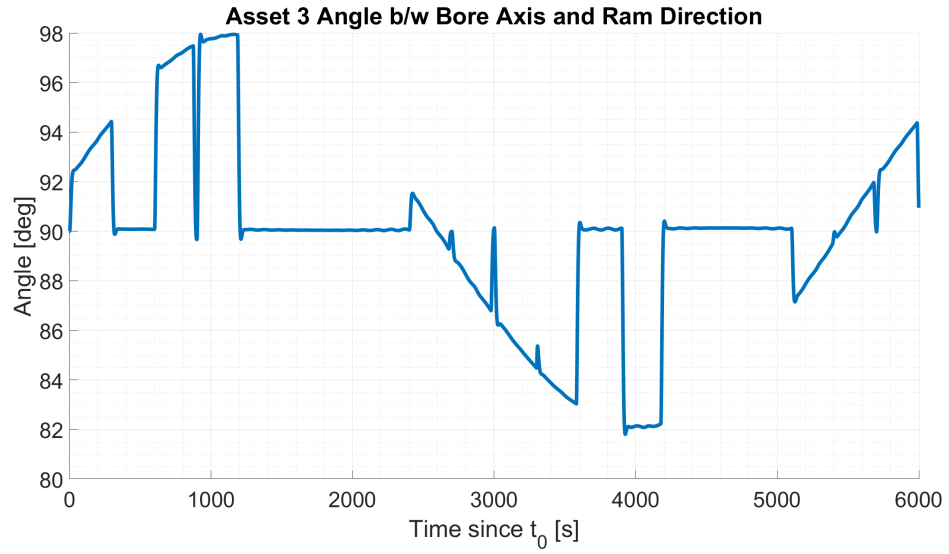


**Figure A.10: Constellation Ram Angle Time Series for Satellite 1**

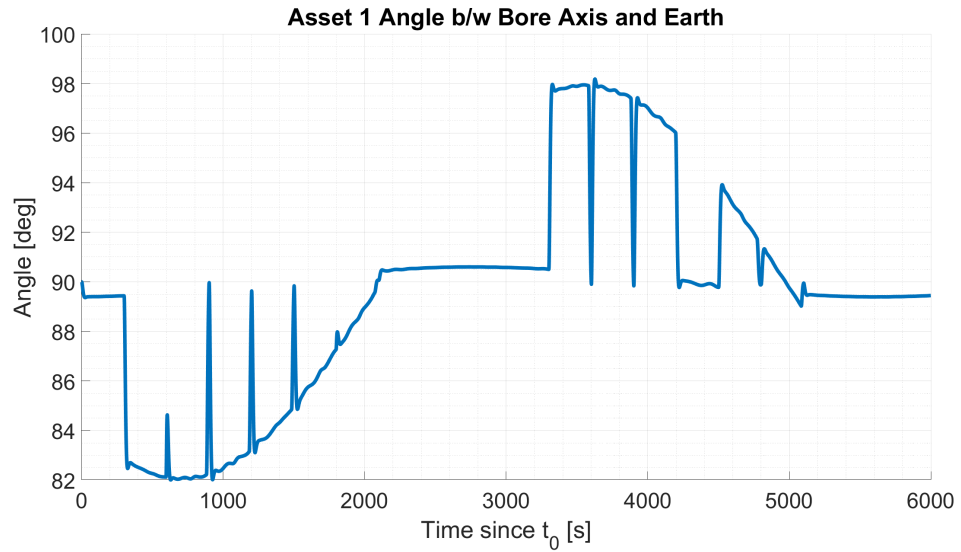


**Figure A.11: Constellation Ram Angle Time Series for Satellite 2**

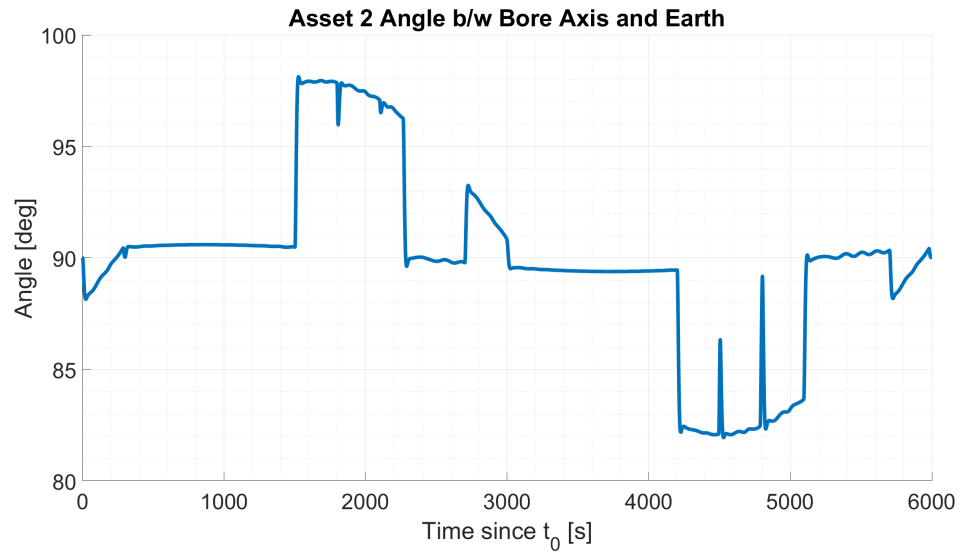




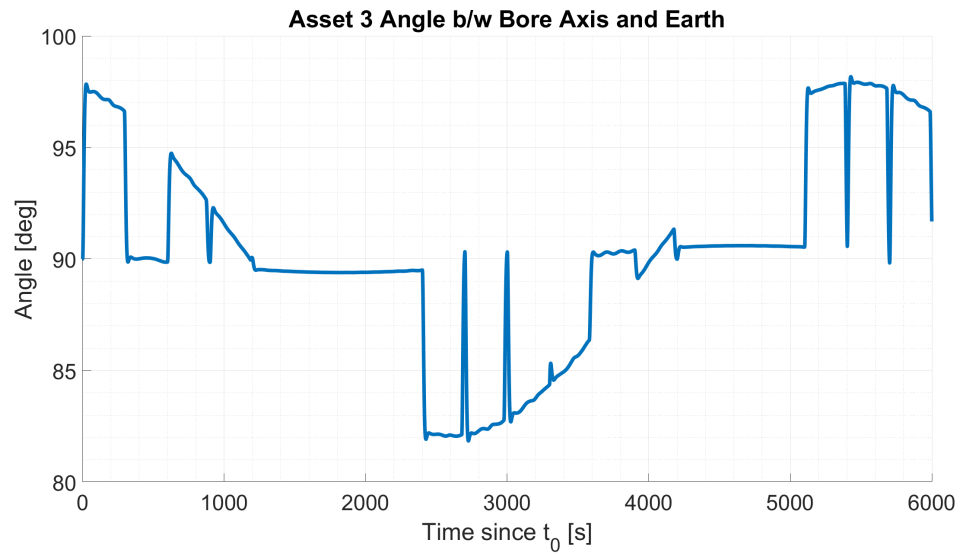
**Figure A.12: Constellation Ram Angle Time Series for Satellite 3**



**Figure A.13: Constellation Nadir Angle Time Series for Satellite 1**



**Figure A.14: Constellation Nadir Angle Time Series for Satellite 2**



**Figure A.15: Constellation Nadir Angle Time Series for Satellite 3**

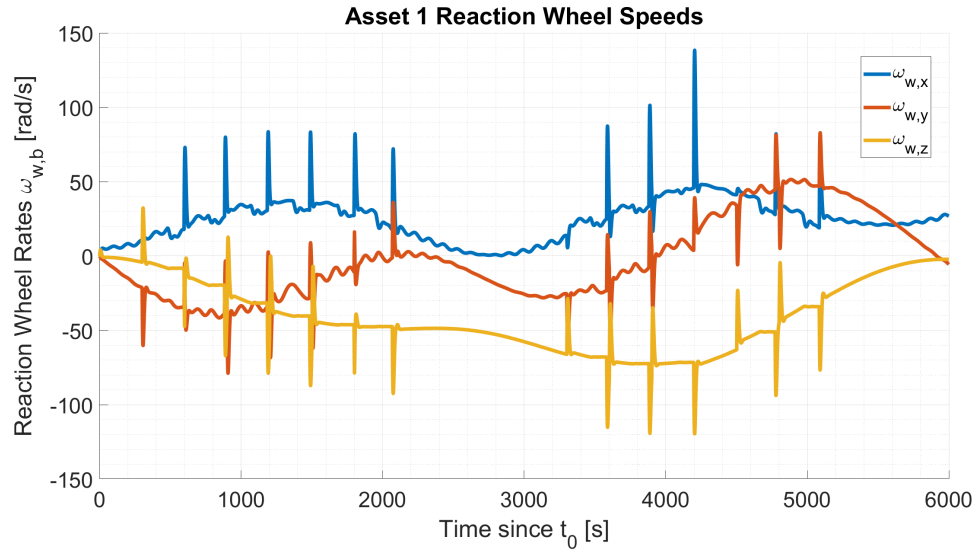


Figure A.16: Constellation Wheel Speed Time Series for Satellite 1

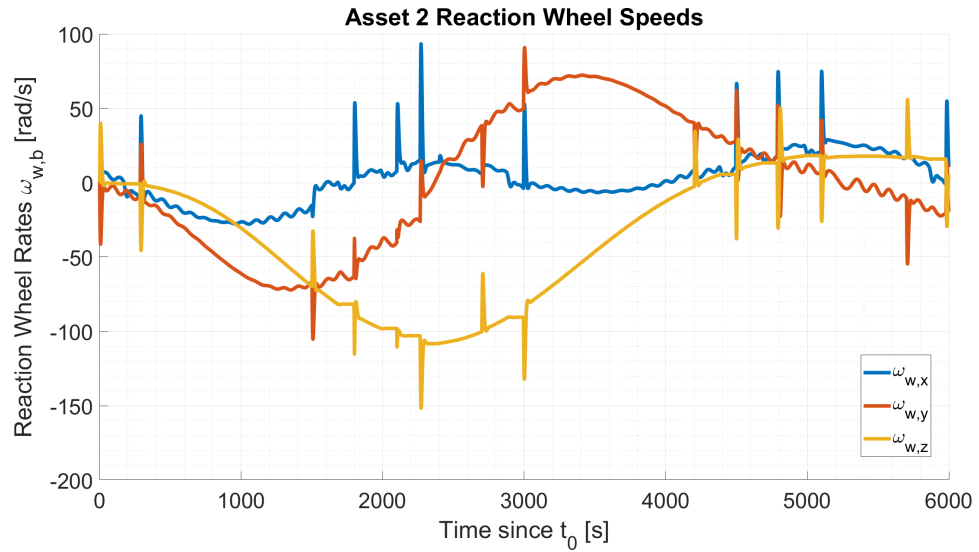


Figure A.17: Constellation Wheel Speed Time Series for Satellite 2

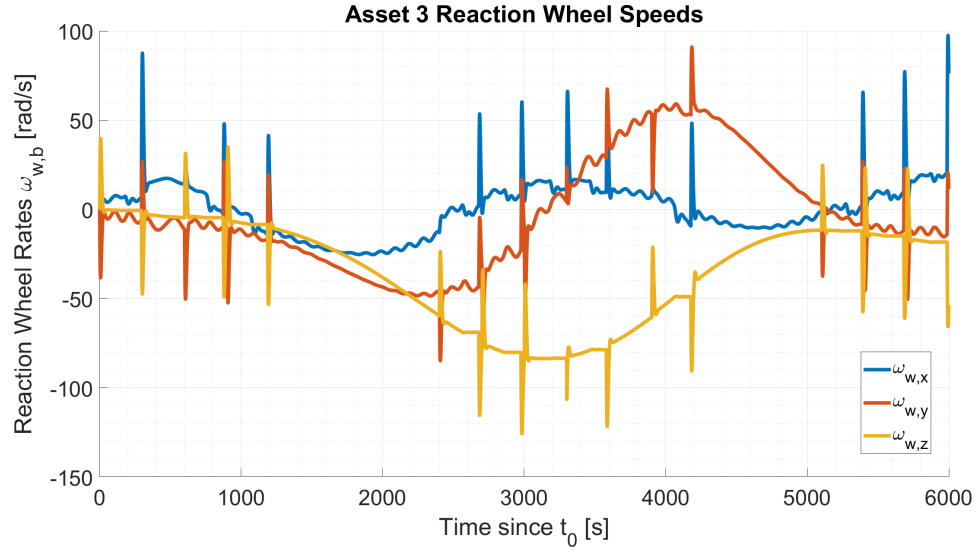


Figure A.18: Constellation Wheel Speed Time Series for Satellite 3

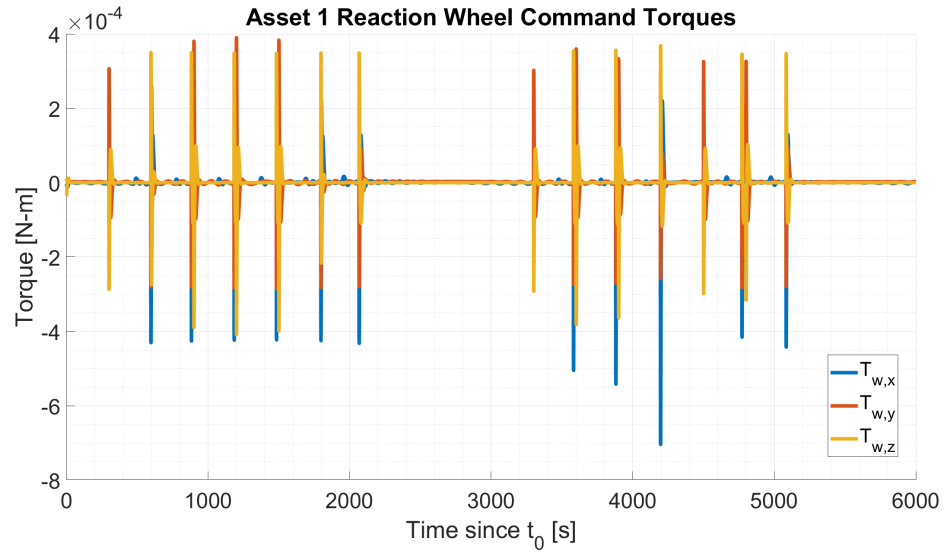


Figure A.19: Constellation Command Torque Series for Satellite 1

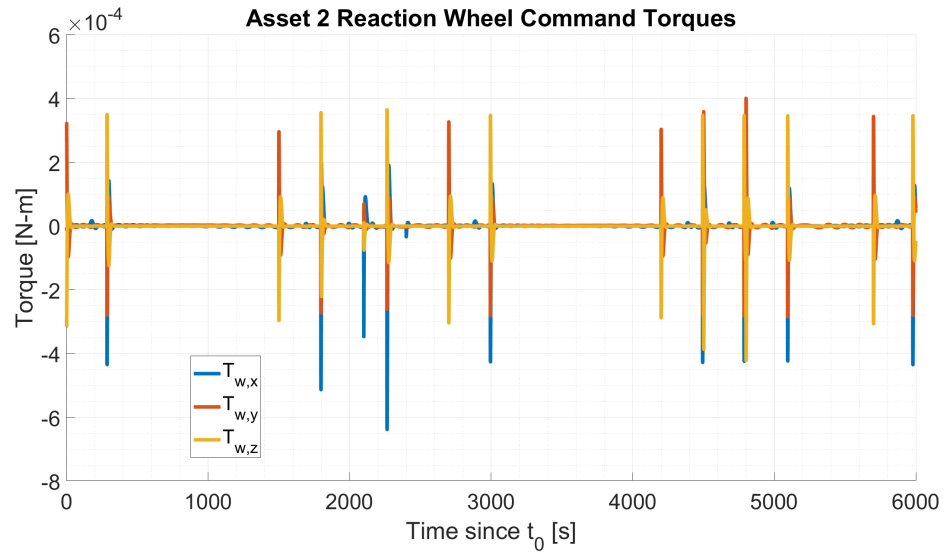


Figure A.20: Constellation Command Torque Series for Satellite 2

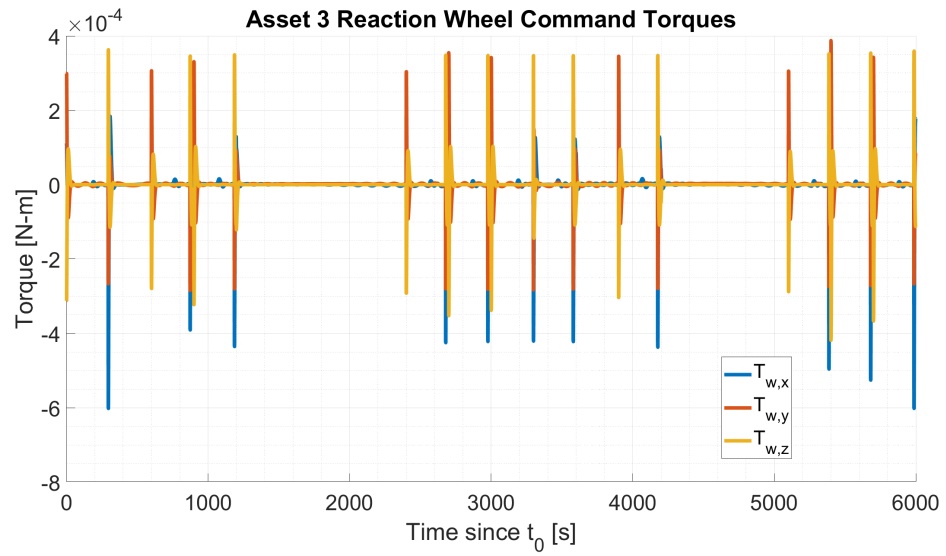
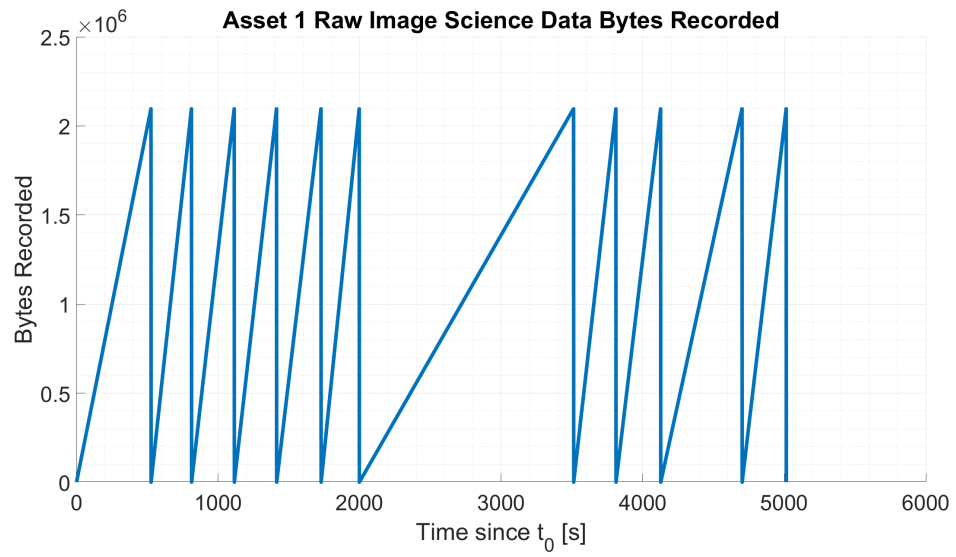
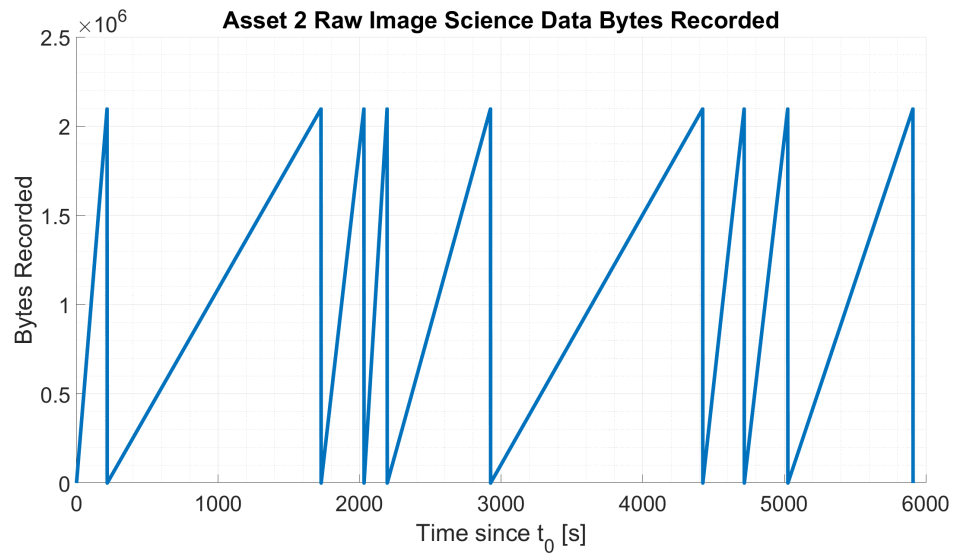


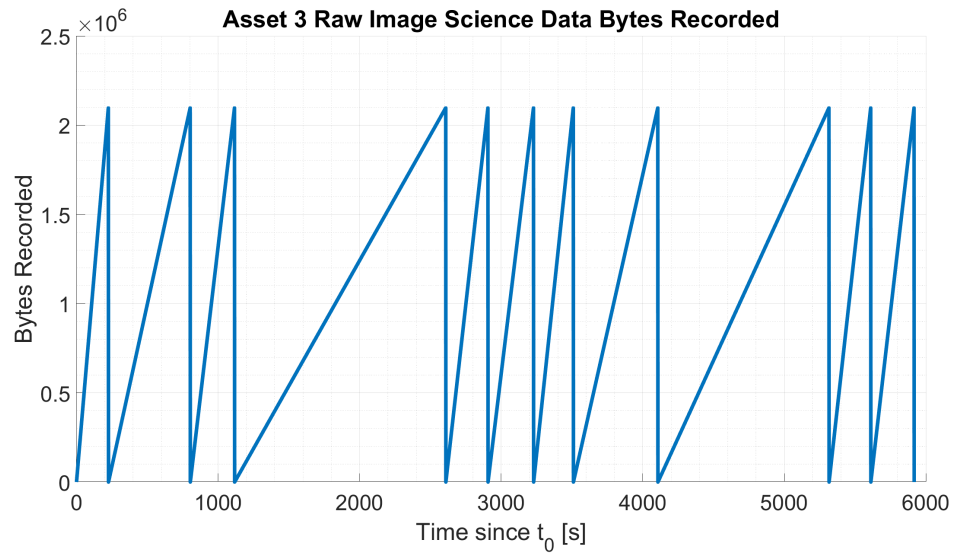
Figure A.21: Constellation Command Torque Series for Satellite 3



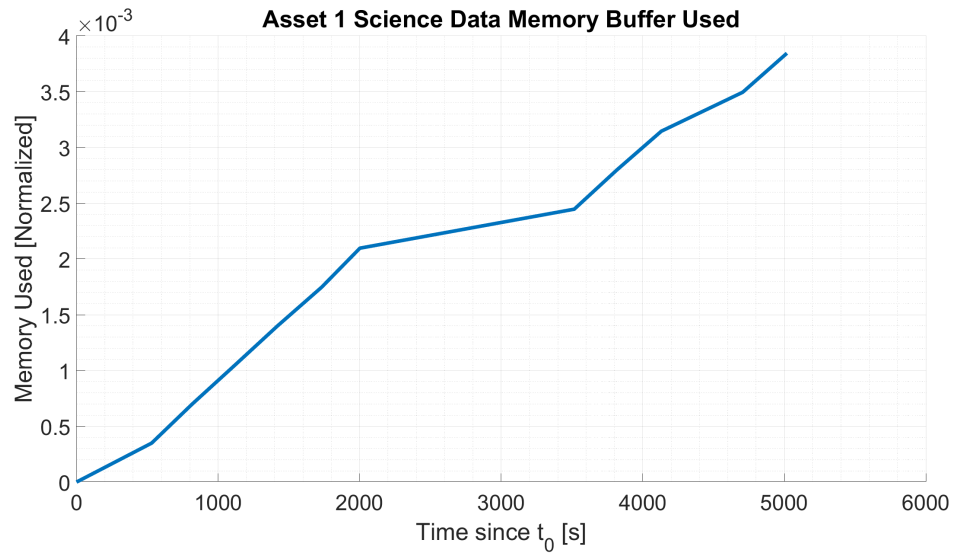
**Figure A.22: Constellation Science Data Recorded Time Series for Satellite 1**



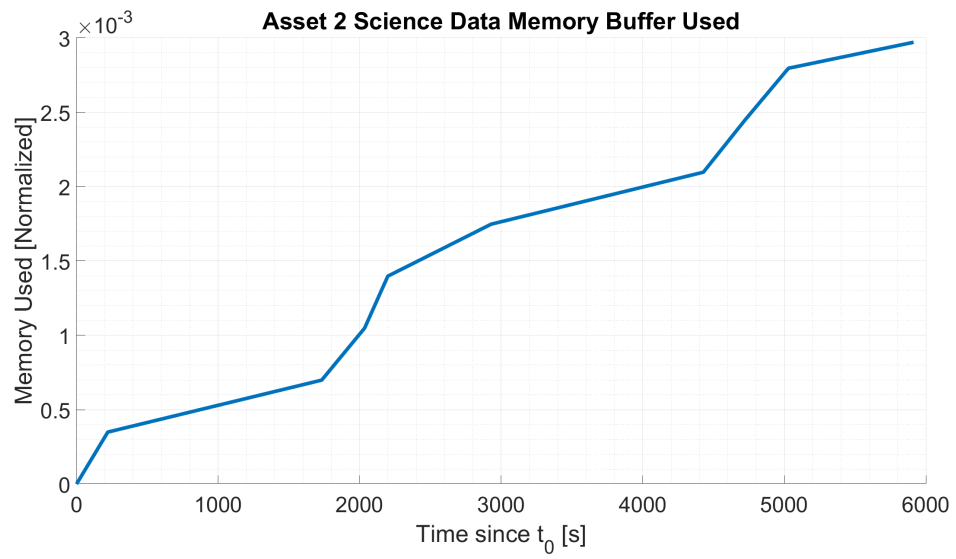
**Figure A.23: Constellation Science Data Recorded Time Series for Satellite 2**



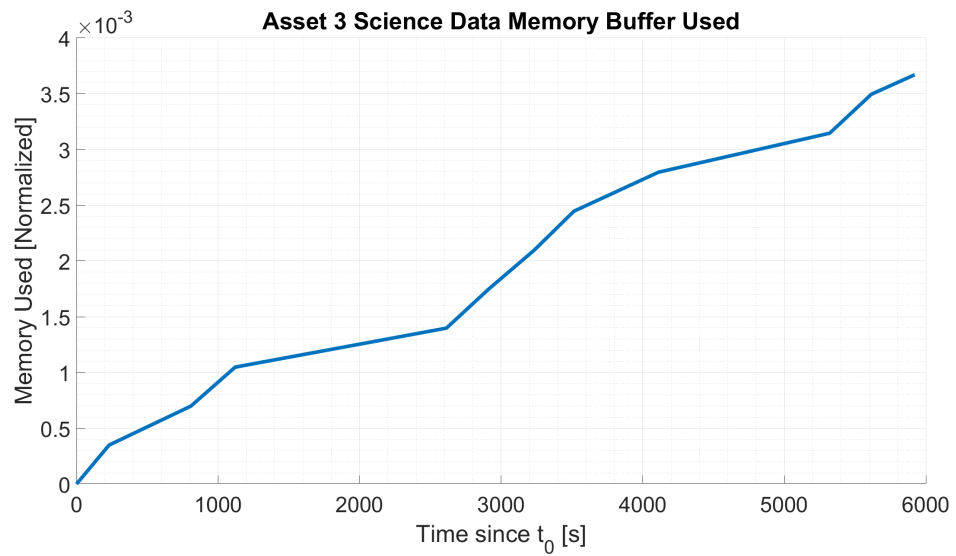
**Figure A.24: Constellation Science Data Recorded Time Series for Satellite 3**



**Figure A.25: Constellation Data Storage Time Series for Satellite 1**

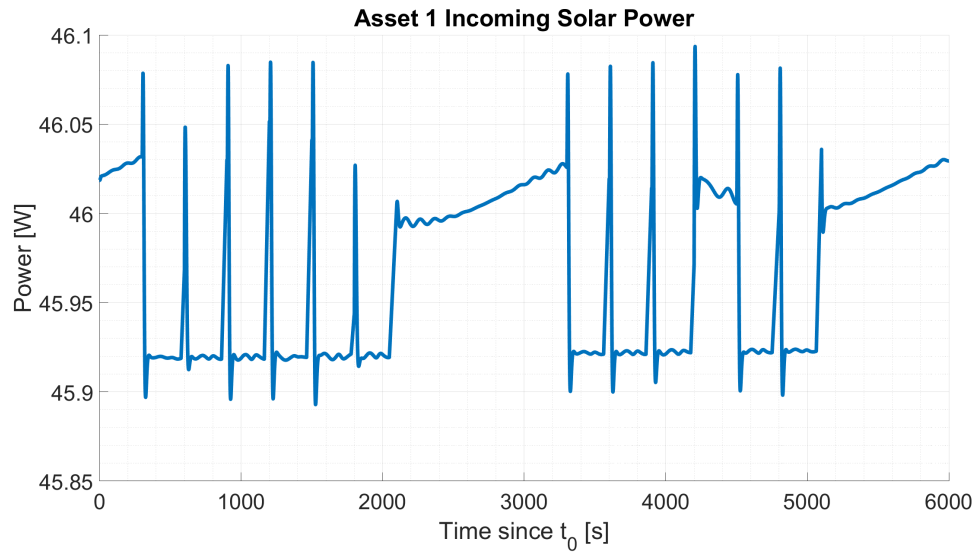


**Figure A.26: Constellation Data Storage Time Series for Satellite 2**

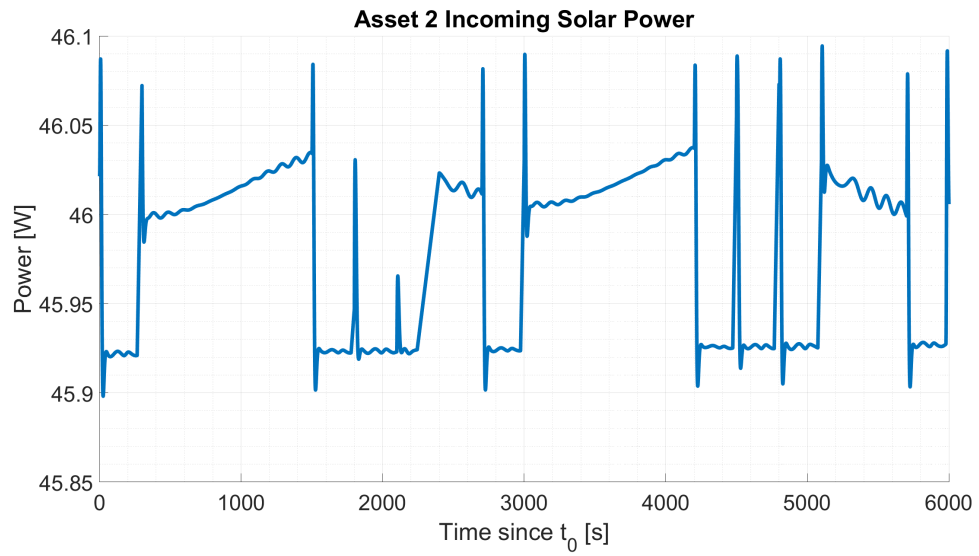


**Figure A.27: Constellation Data Storage Time Series for Satellite 3**





**Figure A.28: Constellation Power Generation Time Series for Satellite 1**



**Figure A.29: Constellation Power Generation Time Series for Satellite 2**

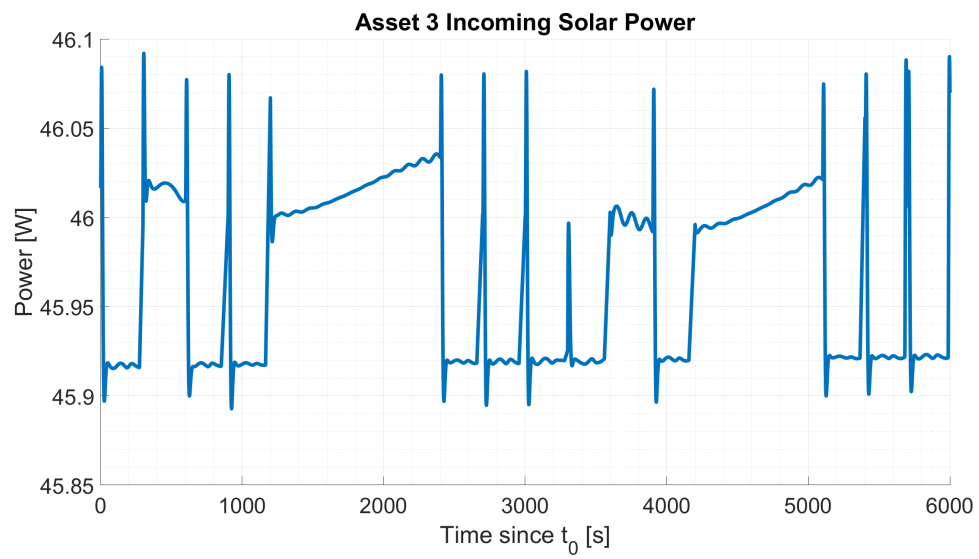


Figure A.30: Constellation Power Generation Time Series for Satellite 3

Appendix B

EXPONENTIAL ATMOSPHERE MODEL

**Table B.1: Exponential Atmosphere Model Coefficients**

$h[km]$	$\rho_0[kg/m^3]$	$h_0[km]$	$h[km]$	$\rho_0[kg/m^3]$	$h_0[km]$
0	1.225	7.249	150	2.070E-9	22.523
25	3.899E-2	6.349	180	5.464E-10	29.740
30	1.774E-2	6.682	200	2.789E-10	37.105
40	3.972E-3	7.554)	250	7.248E-11	45.546
50	1.057E-3	8.382	300	2.418E-11	53.628
60	3.206E-4	7.714	350	9.518E-12	53.298
70	8.770E-5	6.549	400	3.725E-12	58.515
80	1.905E-5	5.799	450	1.585E-12	60.828
90	3.396E-6	5.382	500	6.967E-13	63.822
100	5.297E-7	5.877	600	1.454E-13	71.835
110	9.661E-8	7.263	700	3.614E-14	88.667
120	2.438E-8	9.473	800	1.170E-14	124.64
130	8.484E-9	12.636	900	5.245E-15	181.05
140	3.845E-9	16.149	1000	3.019E-15	268.00

Appendix C

WORLD MAGNETIC MODEL

**Table C.1: WMM Coefficients A**

$n$	$m$	$g_n^m(t_0)$	$h_n^m(t_0)$	$\dot{g}_n^m(t_0)$	$\dot{h}_n^m(t_0)$
1	0	-29438.2	0.0	7.0	0.0
1	1	-1493.5	4796.3	9.0	-30.2
2	0	-2444.5	0.0	-11.0	0.0
2	1	3014.7	-2842.4	-6.2	-29.6
2	2	1679.0	-638.8	0.3	-17.3
3	0	1351.8	0.0	2.4	0.0
3	1	-2351.6	-113.7	-5.7	6.5
3	2	1223.6	246.5	2.0	-0.8
3	3	582.3	-537.4	-11.0	-2.0
4	0	907.5	0.0	-0.8	0.0
4	1	814.8	283.3	-0.9	-0.4
4	2	117.8	-188.6	-6.5	5.8
4	3	-335.6	180.7	5.2	3.8
4	4	69.7	-330.0	-4.0	-3.5
5	0	-232.9	0.0	-0.3	0.0
5	1	360.1	46.9	0.6	0.2
5	2	191.7	196.5	-0.8	2.3
5	3	-141.3	-119.9	0.1	-0.0

**Table C.2: WMM Coefficients B**

$n$	$m$	$g_n^m(t_0)$	$h_n^m(t_0)$	$\dot{g}_n^m(t_0)$	$\dot{h}_n^m(t_0)$
5	4	-157.2	16.0	1.2	3.3
5	5	7.7	100.6	1.4	-0.6
6	0	69.4	0.0	-0.8	0.0
6	1	67.7	-20.1	-0.5	0.3
6	2	72.3	32.8	-0.1	-1.5
6	3	-129.1	59.1	1.6	-1.2
6	4	-28.4	-67.1	-1.6	0.4
6	5	13.6	8.1	0.0	0.2
6	6	-70.3	61.9	1.2	1.3
7	0	81.7	0.0	-0.3	0.0
7	1	-75.9	-54.3	-0.2	0.6
7	2	-7.1	-19.5	-0.3	0.5
7	3	52.2	6.0	0.9	-0.8
7	4	15.0	24.5	0.1	-0.2
7	5	9.1	3.5	-0.6	-1.1
7	6	-3.0	-27.7	-0.9	0.1
7	7	5.9	-2.9	0.7	0.2
8	0	24.2	0.0	-0.1	0.0

**Table C.3: WMM Coefficients C**

$n$	$m$	$g_n^m(t_0)$	$h_n^m(t_0)$	$\dot{g}_n^m(t_0)$	$\dot{h}_n^m(t_0)$
8	1	8.9	10.1	0.2	-0.4
8	2	-16.9	-18.3	-0.2	0.6
8	3	-3.1	13.3	0.5	-0.1
8	4	-20.7	-14.5	-0.1	0.6
8	5	13.3	16.2	0.4	-0.2
8	6	11.6	6.0	0.4	-0.5
8	7	-16.3	-9.2	-0.1	0.5
8	8	-2.1	2.4	0.4	0.1
9	0	5.5	0.0	-0.1	0.0
9	1	8.8	-21.8	-0.1	-0.3
9	2	3.0	10.7	-0.0	0.1
9	3	-3.2	11.8	0.4	-0.4
9	4	0.6	-6.8	-0.4	0.3
9	5	-13.2	-6.9	0.0	0.1
9	6	-0.1	7.9	0.3	-0.0
9	7	8.7	1.0	0.0	-0.1
9	8	-9.1	-3.9	-0.0	0.5
9	9	-10.4	8.5	-0.3	0.2

**Table C.4: WMM Coefficients D**

$n$	$m$	$g_n^m(t_0)$	$h_n^m(t_0)$	$\dot{g}_n^m(t_0)$	$\dot{h}_n^m(t_0)$
10	0	-2.0	0.0	0.0	0.0
10	1	-6.1	3.3	-0.0	0.0
10	2	0.2	-0.4	-0.1	0.1
10	3	0.6	4.6	0.2	-0.2
10	4	-0.5	4.4	-0.1	0.1
10	5	1.8	-7.9	-0.2	-0.1
10	6	-0.7	-0.6	-0.0	0.1
10	7	2.2	-4.2	-0.1	-0.0
10	8	2.4	-2.9	-0.2	-0.1
10	9	-1.8	-1.1	-0.1	0.2
10	10	-3.6	-8.8	-0.0	-0.0
11	0	3.0	0.0	-0.0	0.0
11	1	-1.4	-0.0	0.0	0.0
11	2	-2.3	2.1	-0.0	0.1
11	3	2.1	-0.6	0.0	0.0
11	4	-0.8	-1.1	-0.0	0.1
11	5	0.6	0.7	-0.1	-0.0
11	6	-0.7	-0.2	0.0	-0.0

**Table C.5: WMM Coefficients E**

$n$	$m$	$g_n^m(t_0)$	$h_n^m(t_0)$	$\dot{g}_n^m(t_0)$	$\dot{h}_n^m(t_0)$
11	7	0.1	-2.1	-0.0	0.1
11	8	1.7	-1.5	-0.0	-0.0
11	9	-0.2	-2.6	-0.1	-0.1
11	10	0.4	-2.0	-0.0	-0.0
11	11	3.5	-2.3	-0.1	-0.1
12	0	-2.0	0.0	0.0	0.0
12	1	-0.1	-1.0	0.0	-0.0
12	2	0.5	0.3	-0.0	0.0
12	3	1.2	1.8	0.0	-0.1
12	4	-0.9	-2.2	-0.1	0.1
12	5	0.9	0.3	-0.0	-0.0
12	6	0.1	0.7	0.0	0.0
12	7	0.6	-0.1	-0.0	-0.0
12	8	-0.4	0.3	0.0	0.0
12	9	-0.5	0.2	-0.0	0.0
12	10	0.2	-0.9	-0.0	-0.0
12	11	-0.9	-0.2	-0.0	0.0
12	12	-0.0	0.8	-0.1	-0.1



## Appendix D

### GEOPOTENTIAL DISTURBANCES

#### D.1 Geopotential Harmonics

Gravitational perturbation acceleration is modeled using the spherical harmonic expansion of the perturbing gravitational potential per the below perturbing gravitational potential function [42]:

$$\phi_p(\mathbf{r}) = \frac{\mu}{\|\mathbf{r}\|} \left[ - \sum_{n=2}^{\infty} \left[ J_n \left( \frac{R_e}{\|\mathbf{r}\|} \right)^n P_n \left( \frac{z}{\|\mathbf{r}\|} \right) \right] + \sum_{n=2}^{\infty} \sum_{m=1}^n \left[ \left( \frac{R_e}{\|\mathbf{r}\|} \right)^n P_{n,m} \left( \frac{z}{\|\mathbf{r}\|} \right) (C_{n,m} \cos(m\lambda) + S_{n,m} \sin(m\lambda)) \right] \right] \quad (\text{D.1})$$

$$\tan(\lambda) = \frac{y}{x} \quad (\text{D.2})$$

where  $R_e$  is the equatorial radius, which is 6378.137 kilometers for Earth, and  $\mu$  is the gravitational parameter, equal to the product of the gravitational constant and the mass of the central body, which is equal to 398600.4418 cubic kilometers per seconds squared for the Earth.  $P_n(x)$  are the Legendre polynomials defined as

$$\begin{aligned} P_0(x) &= 1 \\ P_1(x) &= x \\ P_{n+1}(x) &= \frac{(2n+1)xP_n(x) - nP_{n-1}(x)}{n+1} \end{aligned} \quad (\text{D.3})$$

and  $P_{n,m}(x)$  are the associated Legendre functions defined as

$$P_{n,m}(x) = (1-x^2)^{m/2} \frac{d^m}{dx^m} P_n(x) \quad (\text{D.4})$$

$z$  is the z-component of inertial position  $\mathbf{r}$  such that  $z = \mathbf{r} \cdot \hat{\mathbf{z}}_0$ , and  $J_n$ ,  $C_{n,m}$ , and  $S_{n,m}$  are empirically derived constants representing the zonal harmonic coefficients and normalized Stokes coefficients [61] of the geopotential function.

For a central body which is axisymmetric about the z-axis, the Stokes coefficients  $C_{n,m}$  and  $S_{n,m}$  are equal to zero, and the geopotential function simplifies to a sum of the zonal harmonic terms as shown below:

$$\phi_p(\mathbf{r}) = -\frac{\mu}{\|\mathbf{r}\|} \sum_{n=2}^{\infty} J_n \left( \frac{R_e}{\|\mathbf{r}\|} \right)^n P_n \left( \frac{z}{\|\mathbf{r}\|} \right) \quad (\text{D.5})$$

Assuming that oblateness effects of the Earth are the dominant disturbing perturbation due to the Earth's gravitational field, higher order perturbations than those due to  $J_2$  can be calculated for an object in Earth's orbit. A summary of the calculations used for third, fourth, and fifth order zonal harmonic disturbances is presented in the following sections.

## D.2 Third Order Zonal Harmonic

The perturbational acceleration due to the  $J_3$  zonal harmonic can be calculated as follows [61]:

$$\mathbf{r} = \mathbf{r}_0 \quad (\text{D.6})$$

$$r = \|\mathbf{r}\| \quad (\text{D.7})$$

$$\hat{\mathbf{r}} = \frac{\mathbf{r}}{r} \quad (\text{D.8})$$

$$z_r = \frac{z}{r} \quad (\text{D.9})$$

$$\mu_r = \frac{\mu}{r^2} \quad (\text{D.10})$$

$$\gamma_3 = -\frac{5J_3}{2} \left( \frac{R_e}{r} \right)^3 \quad (\text{D.11})$$

$$\mathbf{a}_{J3} = -\gamma_3 \mu_r \left( -\left( 3z_r - 7z_r^3 \right) \hat{\mathbf{r}} + \frac{3}{5} \begin{bmatrix} 0 \\ 0 \\ 1 - 5z_r^2 \end{bmatrix} \right) \quad (\text{D.12})$$

### D.3 Fourth Order Zonal Harmonic

The perturbational acceleration due to the  $J_4$  zonal harmonic can be calculated as follows:

$$x_r = \frac{\hat{\mathbf{x}}_0 \cdot \mathbf{r}}{r} \quad (\text{D.13})$$

$$y_r = \frac{\hat{\mathbf{y}}_0 \cdot \mathbf{r}}{r} \quad (\text{D.14})$$

$$\gamma_4 = -\frac{5J_4}{8} \left( \frac{R_e}{r} \right)^4 \quad (\text{D.15})$$

$$\mathbf{a}_{J_4} = \gamma_4 \mu_r \begin{bmatrix} (3 - 42z_r^2 + 63z_r^4) x_r \\ (3 - 42z_r^2 + 63z_r^4) y_r \\ (-15 + 70z_r^2 - 63z_r^4) z_r \end{bmatrix} \quad (\text{D.16})$$

### D.4 Fifth Order Zonal Harmonic

The perturbational acceleration due to the  $J_5$  zonal harmonic can be calculated as follows:

$$\gamma_5 = -\frac{J_5}{8} \left( \frac{R_e}{r} \right)^5 \quad (\text{D.17})$$

$$\mathbf{a}_{J_5} = \gamma_5 \mu_r \begin{bmatrix} 3(35z_r - 210z_r^3 + 231z_r^5) x_r \\ 3(35z_r - 210z_r^3 + 231z_r^5) y_r \\ (15 - 315z_r^2 + 945z_r^6) z_r \end{bmatrix} \quad (\text{D.18})$$

### D.5 Zonal Harmonic Coefficients

The zonal harmonic coefficients used for calculating zonal harmonic disturbances from  $J_2$ ,  $J_3$ ,  $J_4$ , and  $J_5$  in Earth orbit are shown in Table D.1.

**Table D.1: Zonal Harmonic Coefficients Up To Fifth Order**

Zonal Harmonic	Coefficient
$J_2$	1.0826e-3
$J_3$	2.533e-6
$J_4$	-1.620e-6
$J_5$	-2.273e-7

## Appendix E

### SIMULATED CONCEPTUAL LINK BUDGET SUMMARY

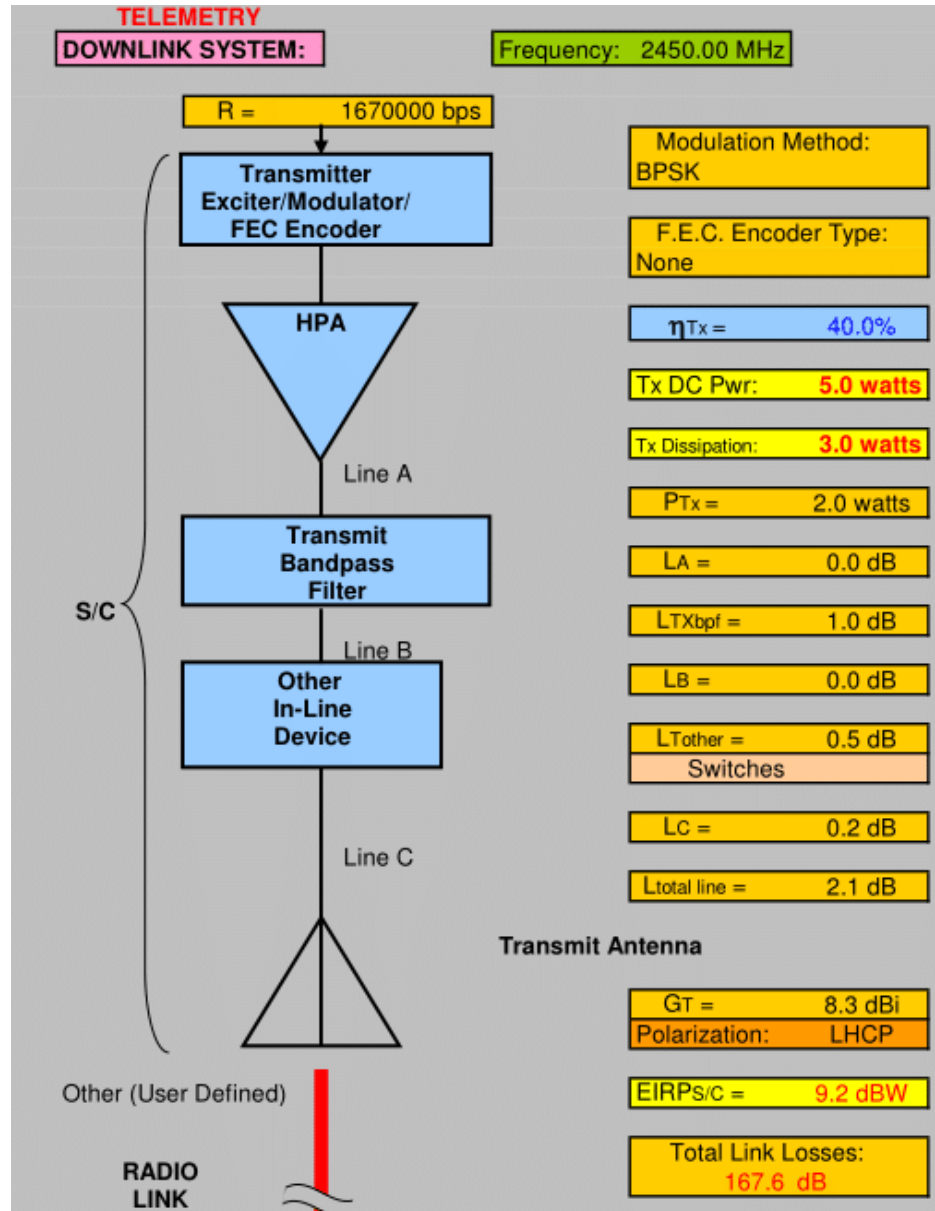


Figure E.1: Simulated Transmitter Downlink Budget

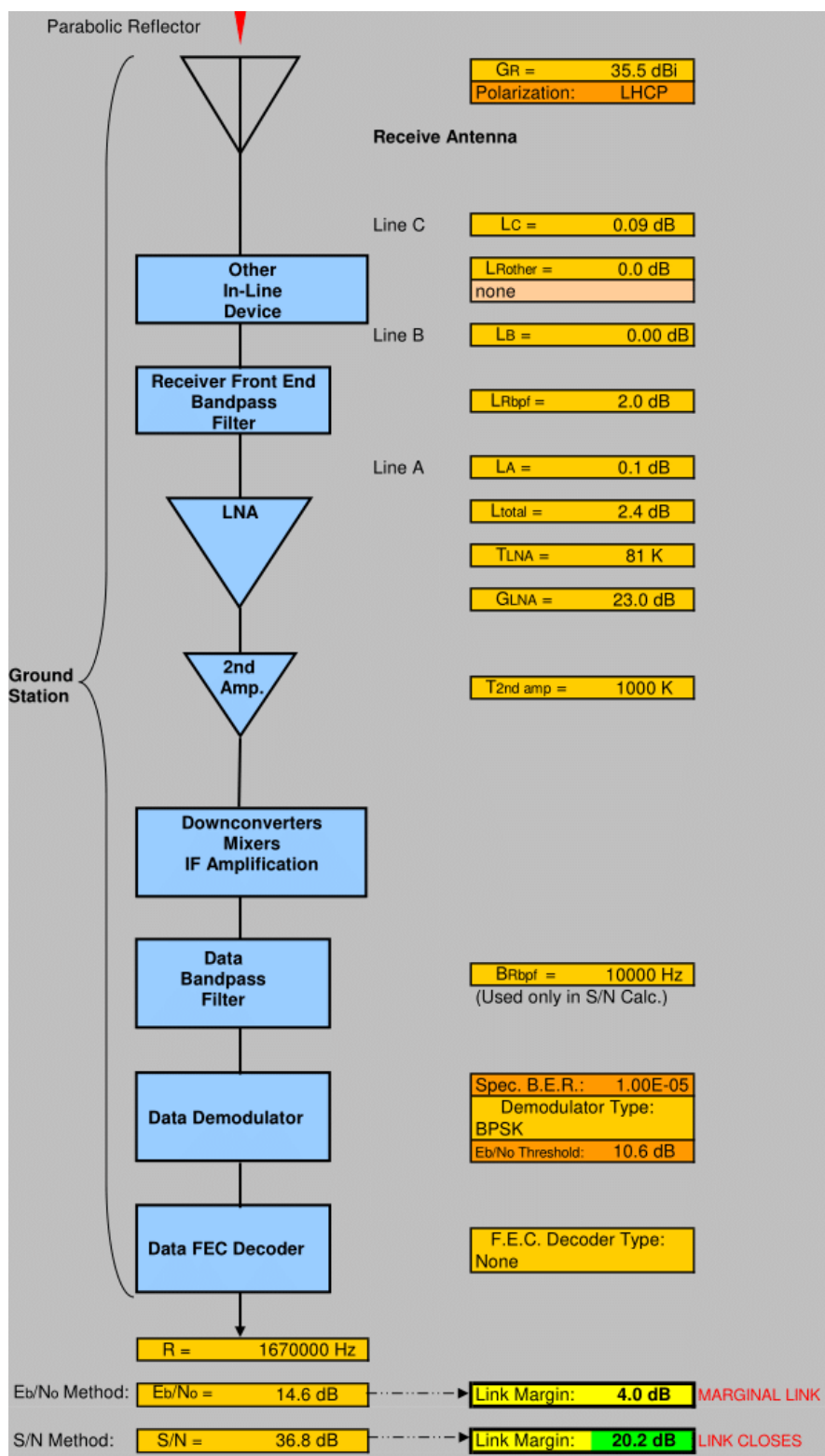


Figure E.2: Simulated Receiver Downlink Budget

## Appendix F

### SIMULATED CONCEPTUAL POWER BUDGET

**Table F.1: Conceptual Power Budget**

Item	#	Avg. Power [W]	Peak Power [W]	Source
IMU	1	1.5	1.5	[4]
Star Tracker	1	1	1	[4]
GPS Module	1	0.5	0.5	[4]
RWP-015	3	0.5	5.5	[51]
iMTQ Board	1	0.175	1.2	[36]
Imaging Payload	1	2	4	[60]
On-board Computer	1	5	5	[4]
XCVR & Antenna	1	5	15	[4]
Thermal Control	1	12	12	[4]
EPS Control System	1	0.5	0.5	Assumed